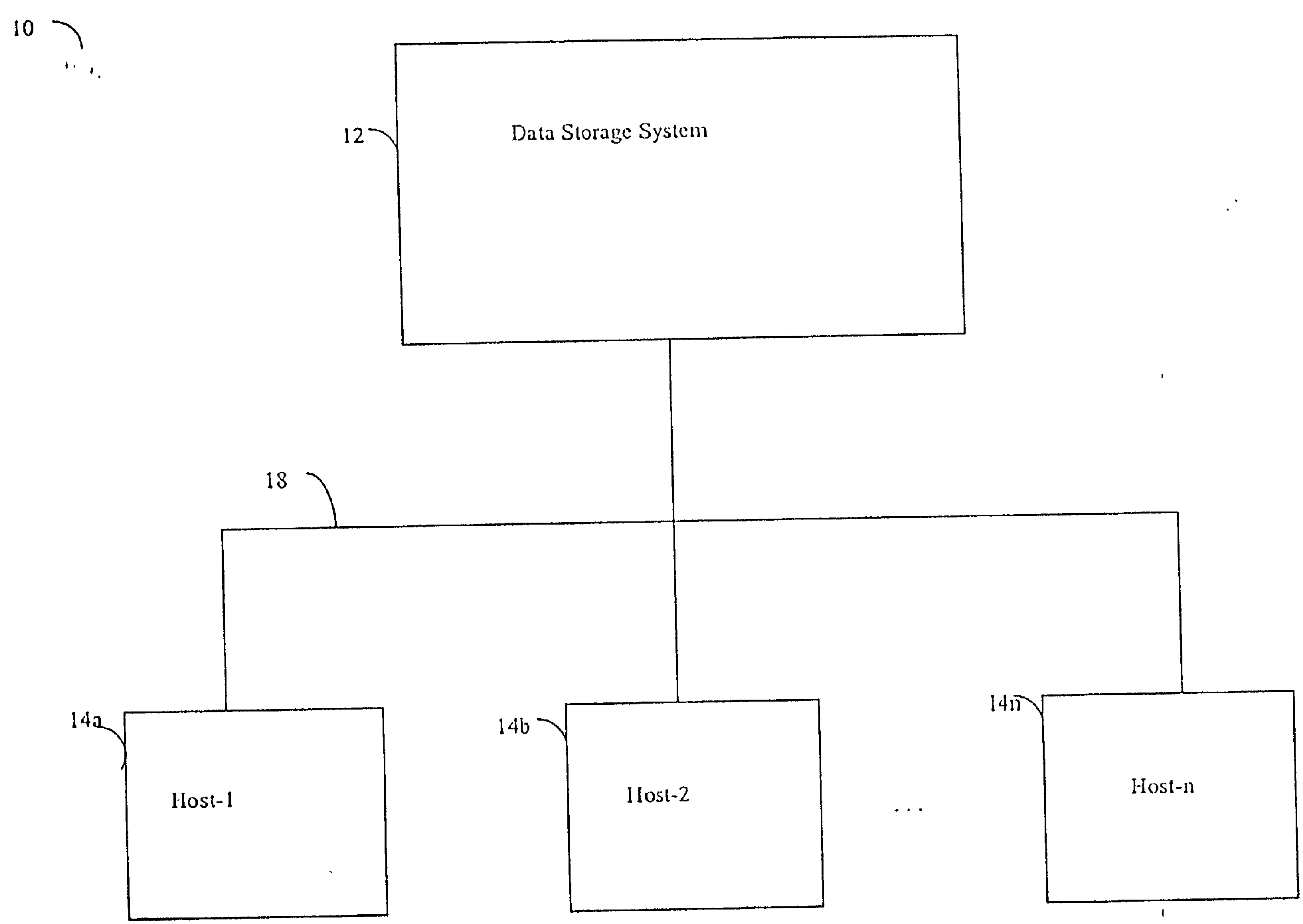


FIGURE 1

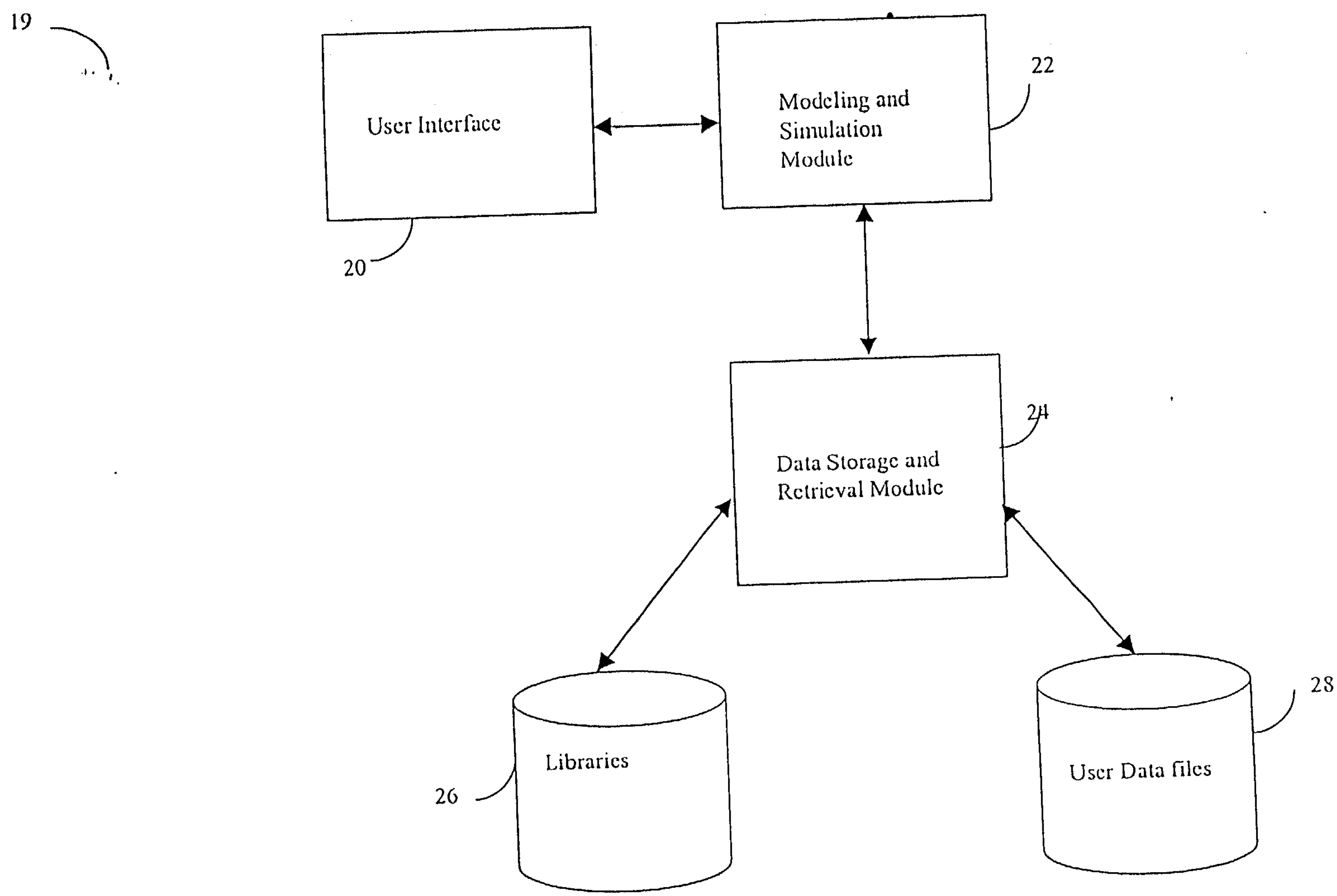

19
User Interface
20
Modeling and Simulation Module
22
Data Storage and Retrieval Module
24
Libraries
26
User Data files
28

FIGURE 2

FIGURE 3

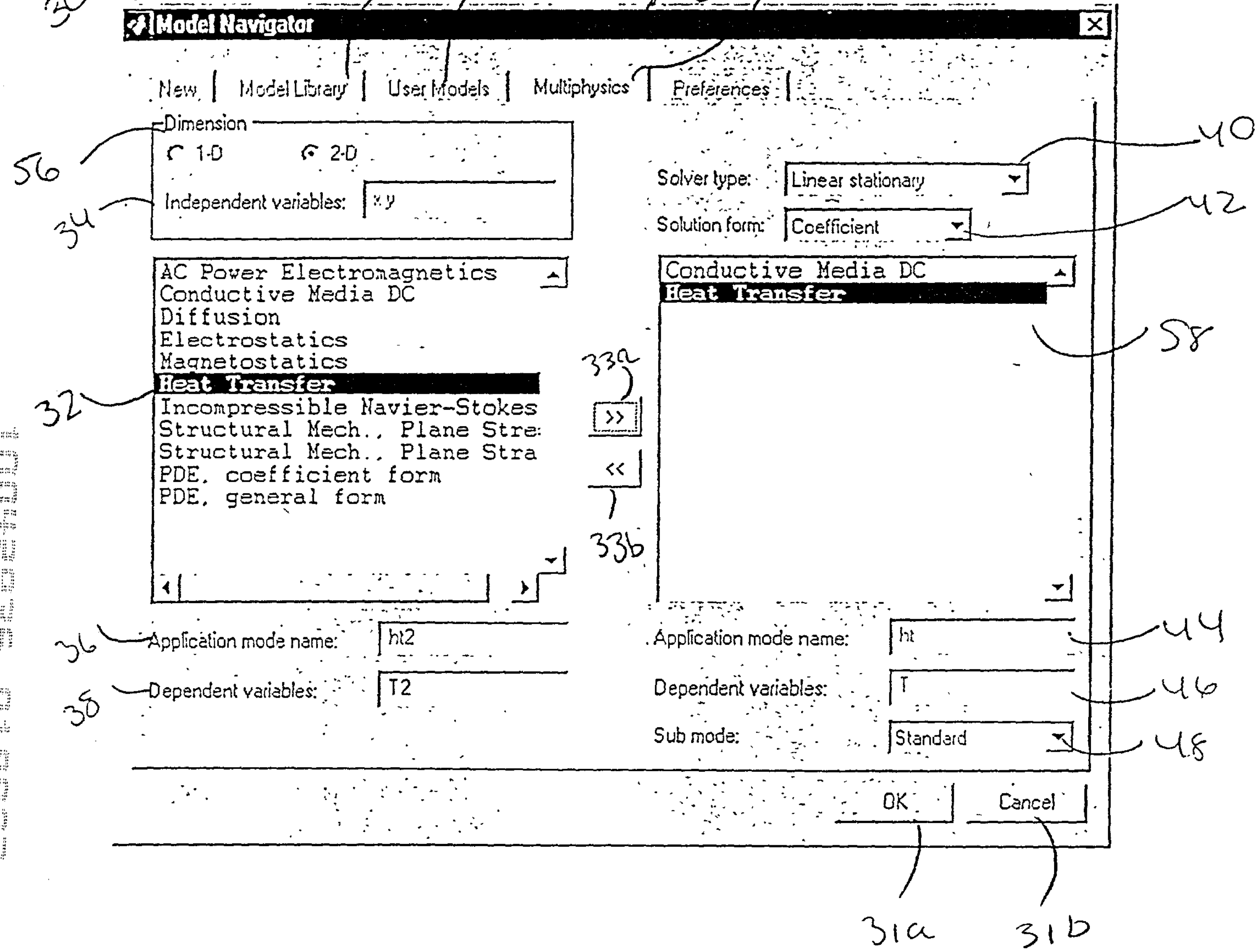

Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Dimension
1-D
2-D
Independent variables:
x y
Solver type:
Linear stationary
Solution form:
Coefficient
AC Power Electromagnetics
Conductive Media DC
Diffusion
Electrostatics
Magnetostatics
Heat Transfer
Incompressible Navier-Stokes
Structural Mech., Plane Stre
Structural Mech., Plane Stra
PDE, coefficient form
PDE, general form
>>
<<
Conductive Media DC
Heat Transfer
Application mode name:
ht2
Dependent variables:
T2
Application mode name:
ht
Dependent variables:
T
Sub mode:
Standard
OK
Cancel
30
31a
31b
32
33a
33b
34
36
38
40
42
44
46
48
50
52
54
56
58

FIGURE ~~2~~ 4

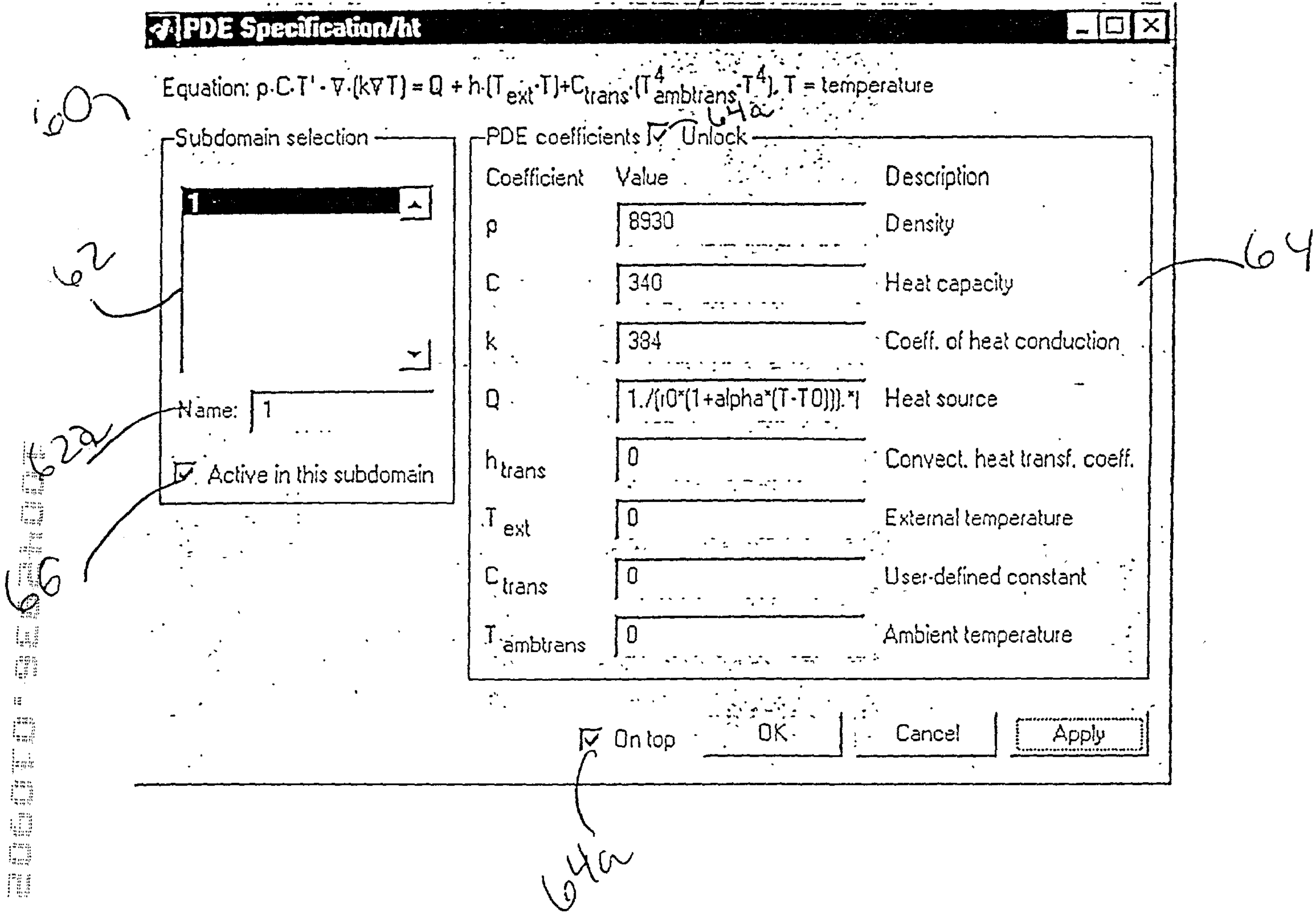
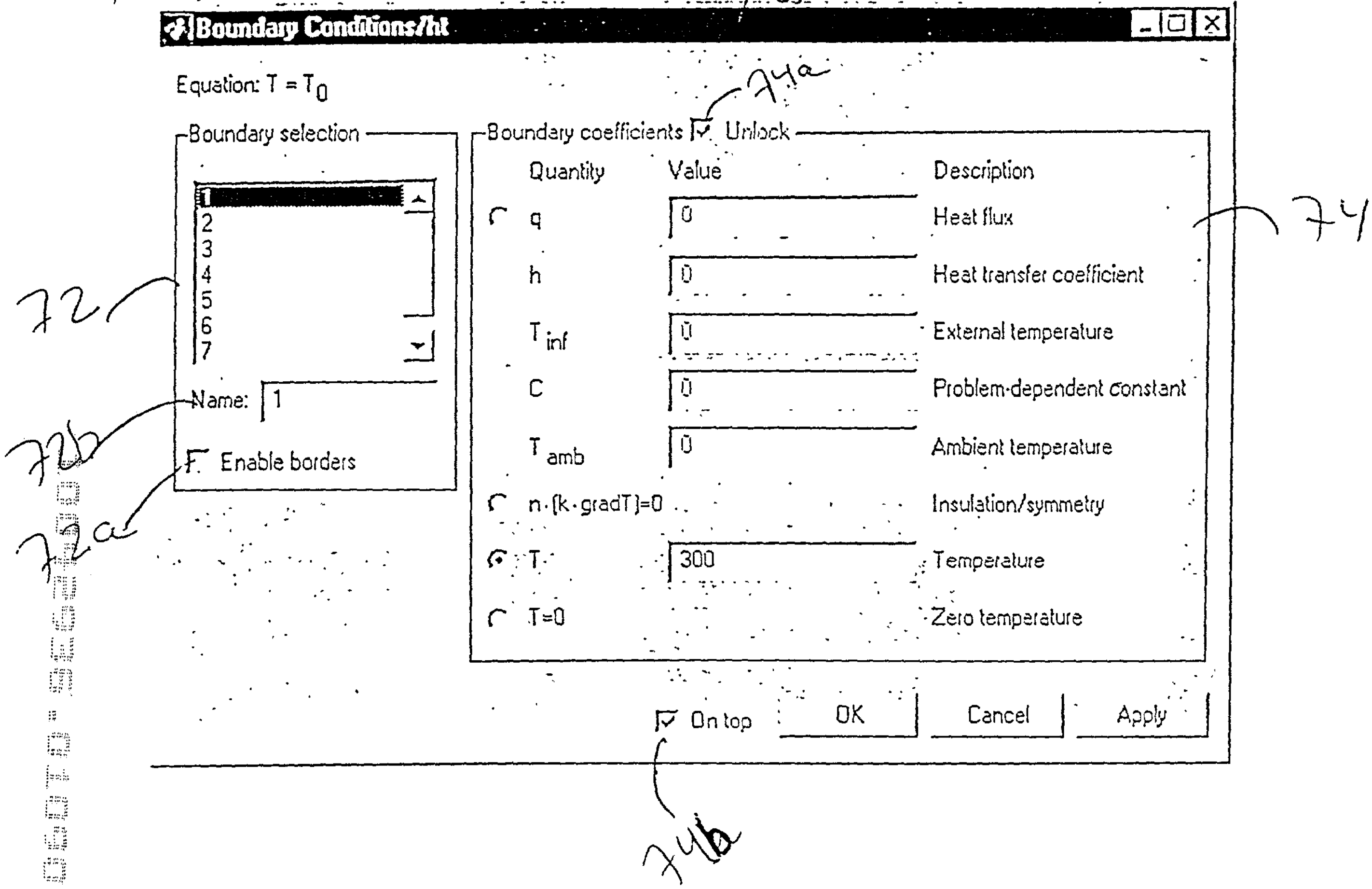
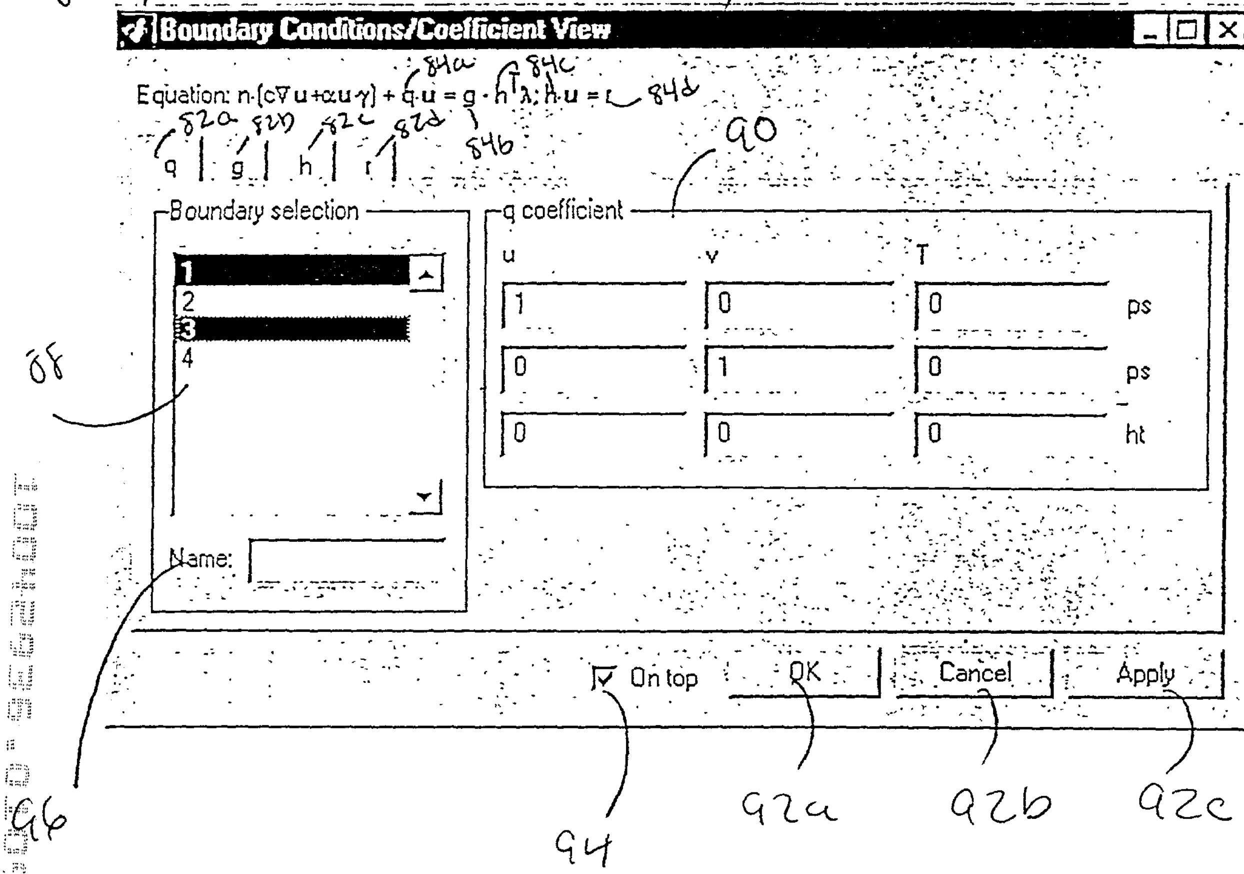

PDE Specification/ht
Equation: ρ·C·T' - ∇·(k∇T) = Q + h·(T ext - T)+C trans·(T⁴ ambtrans - T⁴), T = temperature
60
Subdomain selection
1
62
Name: 1
62a
Active in this subdomain
66
PDE coefficients
Unlock
64a
Coefficient
Value
Description
ρ
8930
Density
C
340
Heat capacity
k
384
Coeff. of heat conduction
Q
1./(r0*(1+alpha*(T-T0))).*I
Heat source
h trans
0
Convect. heat transf. coeff.
T ext
0
External temperature
C trans
0
User-defined constant
T ambtrans
0
Ambient temperature
64
On top
OK
Cancel
Apply
64a

FIGURE ~~3~~ 5



70
Boundary Conditions/ht
Equation: T = T0
Boundary selection
1
2
3
4
5
6
7
Name: 1
Enable borders
72
72b
72a
Boundary coefficients
Unlock
74a
Quantity
Value
Description
q
0
Heat flux
h
0
Heat transfer coefficient
T inf
0
External temperature
C
0
Problem-dependent constant
T amb
0
Ambient temperature
n·(k·gradT)=0
Insulation/symmetry
T
300
Temperature
T=0
Zero temperature
74
On top
OK
Cancel
Apply
74b

FIGURE ~~4~~ 6


80
Boundary Conditions/Coefficient View
Equation: n·(c∇u+αu γ) + q·u = g - h^T λ; h·u = r
84a
84c
84d
84b
82a
82b
82c
82d
q
g
h
r
90
Boundary selection
1
2
3
4
88
Name:
96
q coefficient
u
v
T
1 0 0 ps
0 1 0 ps
0 0 0 ht
On top
OK
Cancel
Apply
94
92a
92b
92c

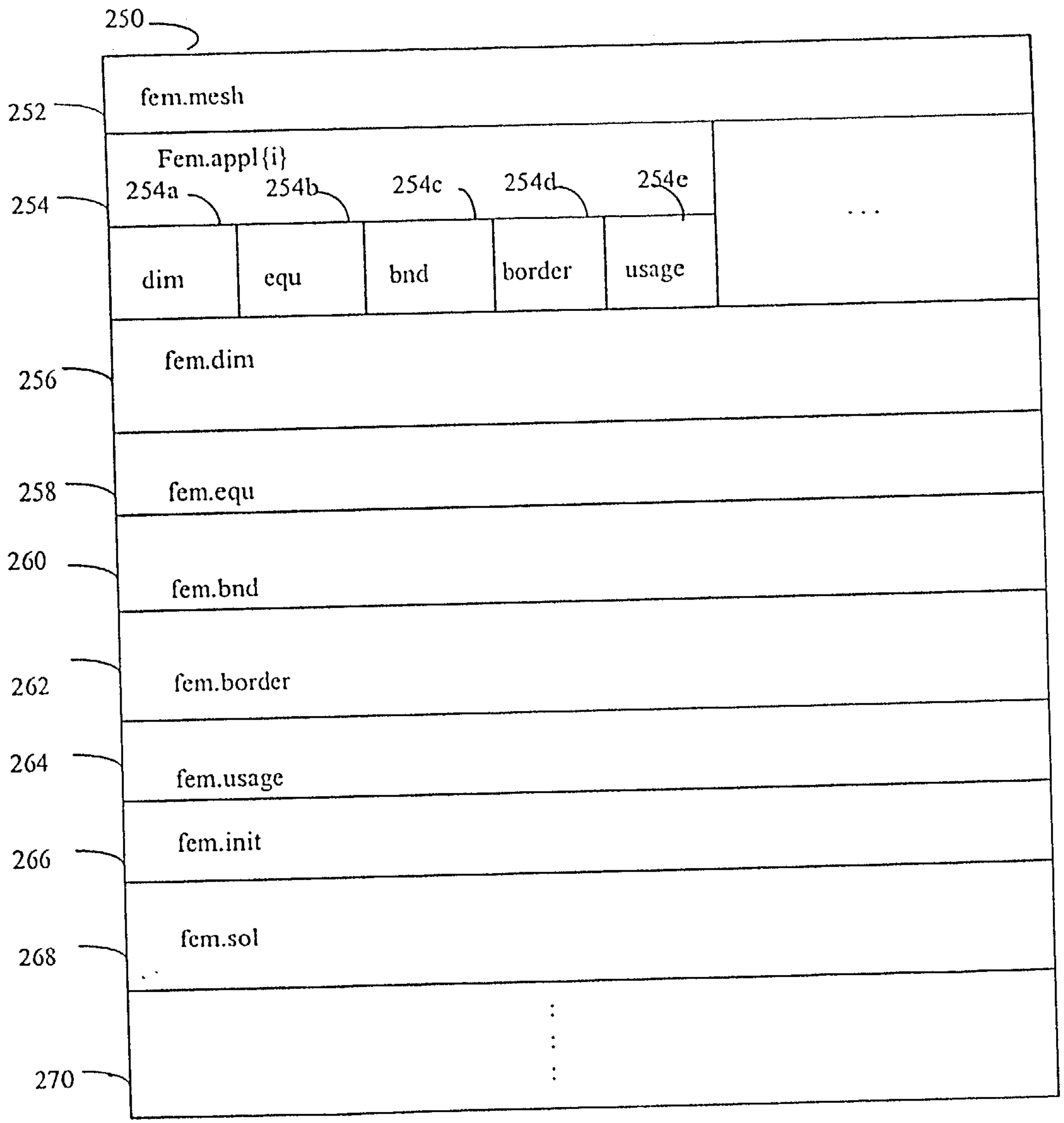

250
252
fem.mesh
254
Fem.appl{i}
254a
254b
254c
254d
254e
dim
equ
bnd
border
usage
...
256
fem.dim
258
fem.equ
260
fem.bnd
262
fem.border
264
fem.usage
266
fem.init
268
fem.sol
270

FIGURE 6A

FIGURE 5/7

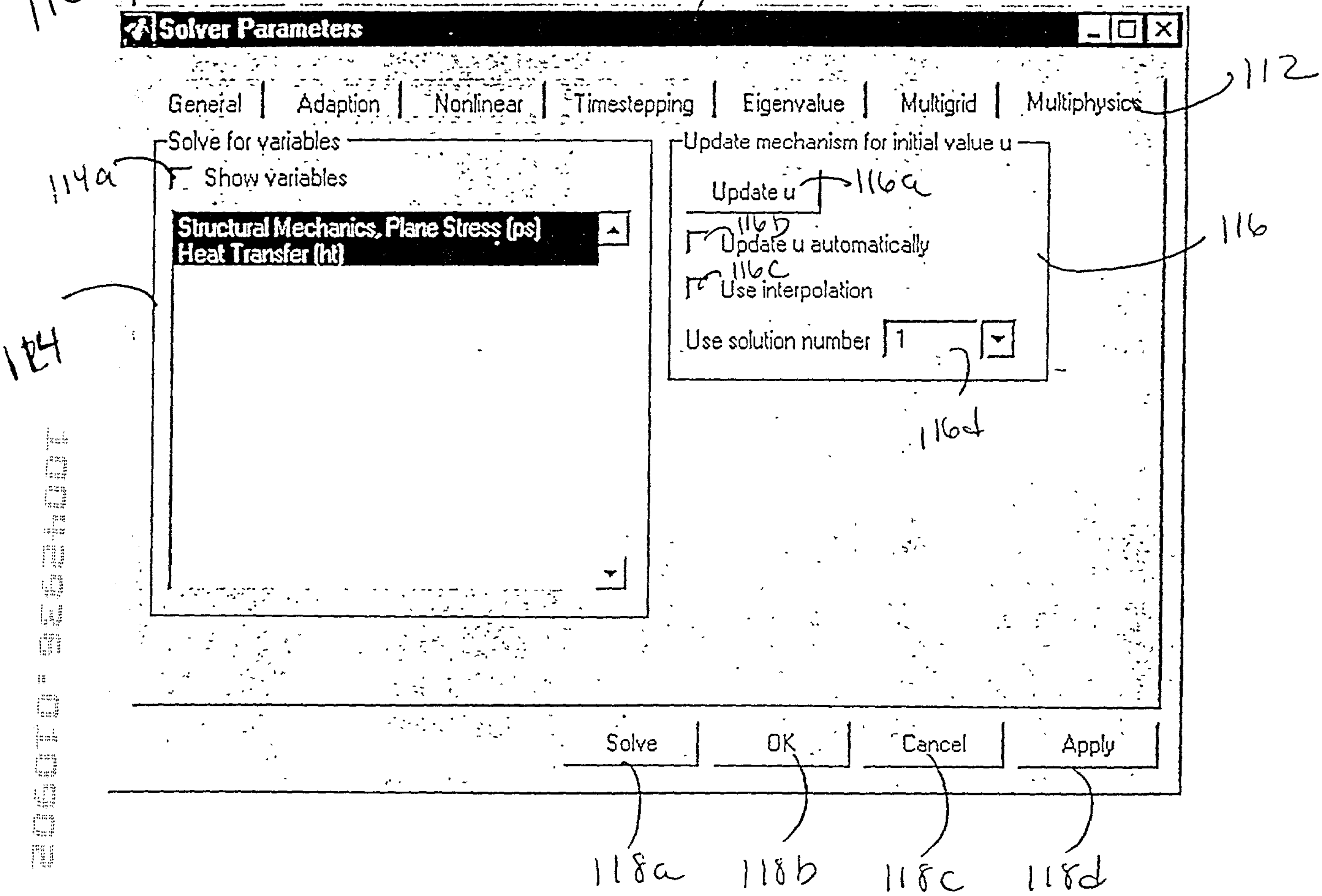

110
Solver Parameters
General
Adaption
Nonlinear
Timestepping
Eigenvalue
Multigrid
Multiphysics
112
Solve for variables
114a
Show variables
Structural Mechanics, Plane Stress (ps)
Heat Transfer (ht)
124
Update mechanism for initial value u
Update u
116a
116b
Update u automatically
116c
Use interpolation
116
Use solution number
1
116d
Solve
OK
Cancel
Apply
118a
118b
118c
118d

FIGURE ~~6~~ 8

$$140\left\{\begin{array}{lll}
d_{a\,lk}\dfrac{\partial u_k}{\partial t}-\dfrac{\partial}{\partial x_j}\left(c_{lkji}\dfrac{\partial u_k}{\partial x_i}+\alpha_{lkj}u_k-\gamma_{lj}\right)+\beta_{lki}\dfrac{\partial u_k}{\partial x_i}+a_{lk}u_k=f_l & \Omega & 142 \\
n_j\left(c_{lkji}\dfrac{\partial u_k}{\partial x_i}+\alpha_{lkj}u_k-\gamma_{lj}\right)+q_{lk}u_k=g_l-h_{ml}\lambda_m & \partial\Omega & 146a \\
h_{ml}u_l=r_m & \partial\Omega & 146b
\end{array}\right\} 146$$

FIGURE ~~7~~ 9

$$150\left\{\begin{array}{lll}
d_{a\,lk}\dfrac{\partial u_k}{\partial t}+\dfrac{\partial \Gamma_{lj}}{\partial x_j}=F_l & \Omega & 152 \\
-n_j\Gamma_{lj}=G_l+\dfrac{\partial R_m}{\partial u_l}\lambda_m & \partial\Omega & 154a \\
0=R_m & \partial\Omega & 154b
\end{array}\right\} 154$$

FIGURE 10

324

$$
\left\{
\begin{array}{ll}
\gamma_{Ij} = \Gamma_{Ij} & f_I = F_I \\[2ex]
c_{IkjI} = -\dfrac{\partial \Gamma_{Ij}}{\partial\left(\dfrac{\partial u_k}{\partial x_I}\right)} & \alpha_{Ikj} = -\dfrac{\partial \Gamma_{Ij}}{\partial u_k} \\[4ex]
\beta_{IkI} = -\dfrac{\partial F_I}{\partial\left(\dfrac{\partial u_k}{\partial x_I}\right)} & a_{Ik} = -\dfrac{\partial F_I}{\partial u_k} \\[4ex]
g_I = G_I & r_I = R_I \\[2ex]
q_{Ik} = -\dfrac{\partial G_I}{\partial u_k} & h_{Ik} = -\dfrac{\partial R_I}{\partial u_k}
\end{array}
\right.
$$

FIGURE ~~0~~ 11

240

$$
\begin{cases}
\Gamma_{lj} = -c_{lkji}\dfrac{\partial u_k}{\partial x_i} - \alpha_{lkj}u_k + \gamma_{lj} \\[2ex]
F_l = f_l - \beta_{lki}\dfrac{\partial u_k}{\partial x_i} - a_{lk}u_k \\[1ex]
G_l = g_l - q_{lk}u_k \\
R_m = r_m - h_{ml}u_l
\end{cases}
$$

FIG 12

$$\left\{\begin{array}{l}
\int_{\Omega}\left(\left(c_{lkji}\dfrac{\partial u_k}{\partial x_i}+\alpha_{lkj}u_k\right)\dfrac{\partial v}{\partial x_j}+\left(d_{a\,lk}\dfrac{\partial u_k}{\partial t}+\beta_{lki}\dfrac{\partial u_k}{\partial x_i}+a_{lk}u_k\right)v\right)dx+ \\
\int_{\partial\Omega} q_{lk}u_k v\,ds = \int_{\Omega}\left(\gamma_{lj}\dfrac{\partial v}{\partial x_j}+f_l v\right)dx+\int_{\partial\Omega}(g_l-h_{ml}\lambda_m)v\,ds \\
\int_{\partial\Omega}\mu h_{mk}u_k\,ds=\int_{\partial\Omega}\mu r_m\,ds
\end{array}\right.$$

FIG 13

$$302\begin{cases} \int_{\Omega}\left(\Gamma_{lj}\frac{\partial v}{\partial x_j}+F_l v-d_{alk}\frac{\partial u_k}{\partial t}v\right)dx+\int_{\partial\Omega}\left(G_l+\frac{\partial R_m}{\partial u_l}\lambda_m\right)v\,ds=0 \\ \int_{\partial\Omega} R_m \mu\, ds = 0 \end{cases}$$

FIG 14

364 $$U_k(x) = \sum_{I=1}^{N_p} U_{I,k}\phi_I(x), \qquad \Lambda_m(x) = \sum_{K=1}^{N_e}\sum_{L=1}^{n} \Lambda_{K,L,m}\Psi_{K,L}(x)$$

FIG 15

306

$$\left\{\begin{aligned}
&\int_{\tau}\left(c_{Ikji}U_{I,k}\frac{\partial\phi_I}{\partial x_i}+\alpha_{Ikj}U_{I,k}\phi_I\right)\frac{\partial\phi_J}{\partial x_j}\,dx+\\
&\int_{\tau}\left(d_{a\,Ik}\frac{\partial U_{I,k}}{\partial t}\phi_I+\beta_{Iki}U_{I,k}\frac{\partial\phi_I}{\partial x_i}+a_{Ik}U_{I,k}\phi_I\right)\phi_J\,dx+\\
&\int_{\partial\tau}q_{Ik}U_{I,k}\phi_I\phi_J\,ds=\int_{\tau}\left(\gamma_{Ij}\frac{\partial\phi_J}{\partial x_j}+f_I\phi_J\right)dx+\\
&\int_{\partial\tau}(g_I-h_{mI}\Lambda_{K,L,m}\Psi_{K,L})\phi_J\,ds
\end{aligned}\right.$$

FIG 16

308

$$\int_{\partial\tau} h_{mk} U_{I,k} \phi_I \Psi_{K,L}\, ds = \int_{\partial\tau} r_m \Psi_{K,L}\, ds$$

FIG 17

312

$$\begin{cases} \int_{\tau}\left(\Gamma_{lj}\frac{\partial\phi_J}{\partial x_j}+F_l\phi_J-d_{alk}\frac{\partial u_k}{\partial t}\phi_J\right)dx+\int_{\partial\tau}\left(G_l+\frac{\partial R_m}{\partial u_l}\Lambda_{K,L,m}\Psi_{K,L}\right)\phi_J\,ds=0 \\ \int_{\partial\tau}R_m\Psi_{K,L}\,ds=0 \end{cases}$$

FIG 18

310

$$DA_{(J,I),(I,k)} = \int_\tau d_{a\,Ik}\phi_I\phi_J\,dx$$

$$C_{(J,I),(I,k)} = \int_\tau c_{Ikji}\frac{\partial\phi_I}{\partial x_i}?\frac{\partial\phi_J}{\partial x_j}\,dx$$

$$AL_{(J,I),(I,k)} = \int_\tau \alpha_{Ikj}\phi_I?\frac{\partial\phi_J}{\partial x_j}\,dx$$

$$BE_{(J,I),(I,k)} = \int_\tau \beta_{Iki}\frac{\partial\phi_I}{\partial x_i}\phi_J\,dx$$

$$A_{(J,I),(I,k)} = \int_\tau a_{Ik}\phi_I\phi_J\,dx$$

$$Q_{(J,I),(I,k)} = \int_{\partial\tau} q_{Ik}\phi_I\phi_J\,ds$$

$$GA_{(J,I)} = \int_\tau \gamma_{Ij}\frac{\partial\phi_J}{\partial x_j}\,dx$$

$$F_{(J,I)} = \int_\tau f_I\phi_J\,dx$$

$$G_{(J,I)} = \int_{\partial\tau} g_I\phi_J\,ds$$

$$H_{(K,L,m),(I,k)} = \int_{\partial\tau} h_{mk}\phi_I\Psi_{K,L}\,ds$$

$$R_{(K,L,m)} = \int_{\partial\tau} r_m\Psi_{K,L}\,ds$$

FIG 19

$$\begin{cases} DA\dfrac{\partial U}{\partial t} + (C + AL + BE + A + Q)U + H^{T}\Lambda = GA + F + G \\ HU = R \end{cases}$$

FIG 20

$$\left\{\begin{array}{l} DA\dfrac{\partial U}{\partial t} + H^T\Lambda = GA + F + G \\ R = 0 \end{array}\right.$$

FIG 21

$$326\left\{\begin{aligned} &J(U^{(k)})\Delta U^{(k)} = -\rho(U^{(k)}) \\ &U^{(k+1)} = U^{(k)} + \lambda_k \Delta U^{(k)} \end{aligned}\right.$$

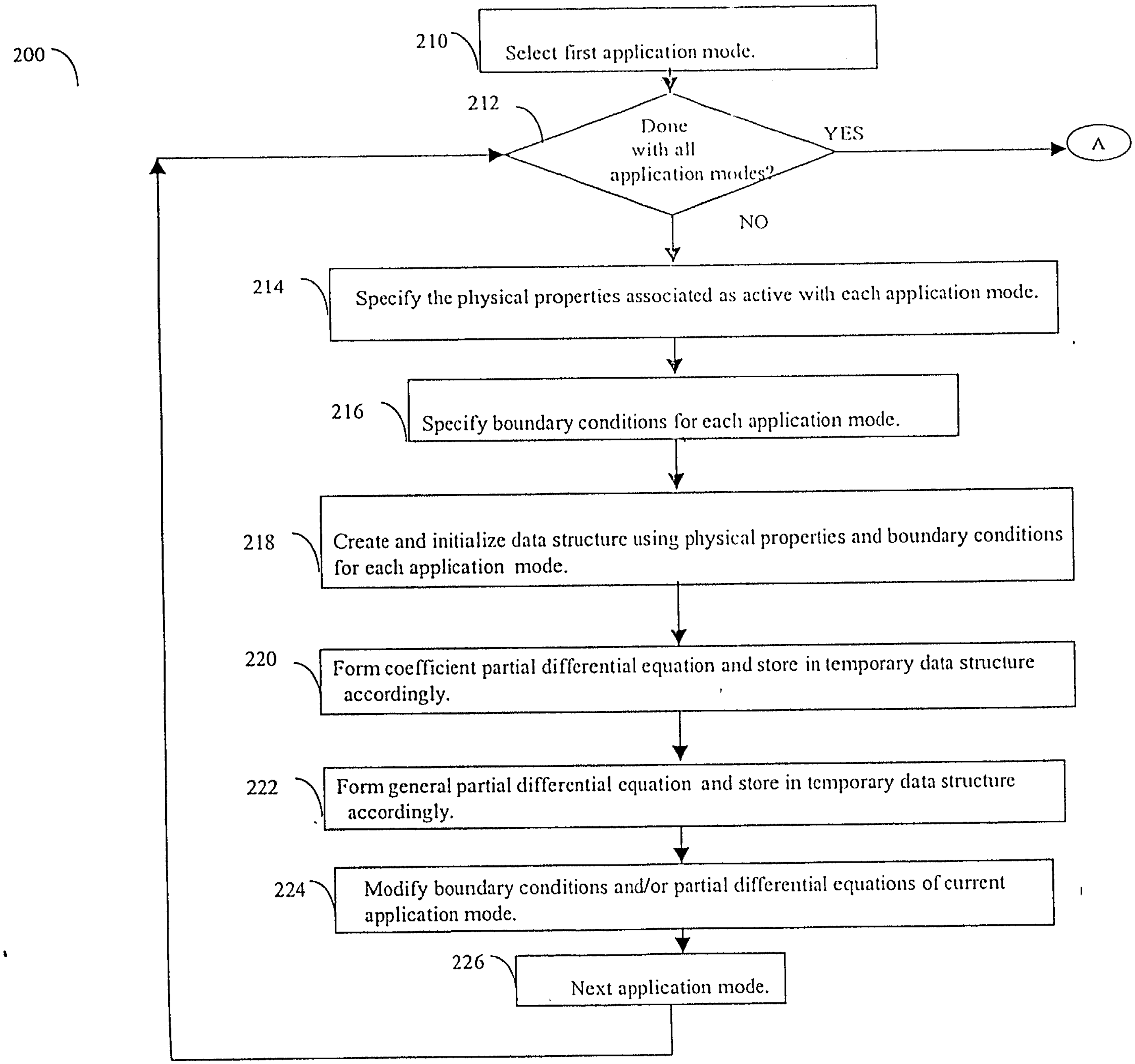

200
210
Select first application mode.
212
Done with all application modes?
YES
A
NO
214
Specify the physical properties associated as active with each application mode.
216
Specify boundary conditions for each application mode.
218
Create and initialize data structure using physical properties and boundary conditions for each application mode.
220
Form coefficient partial differential equation and store in temporary data structure accordingly.
222
Form general partial differential equation and store in temporary data structure accordingly.
224
Modify boundary conditions and/or partial differential equations of current application mode.
226
Next application mode.

FIGURE 22

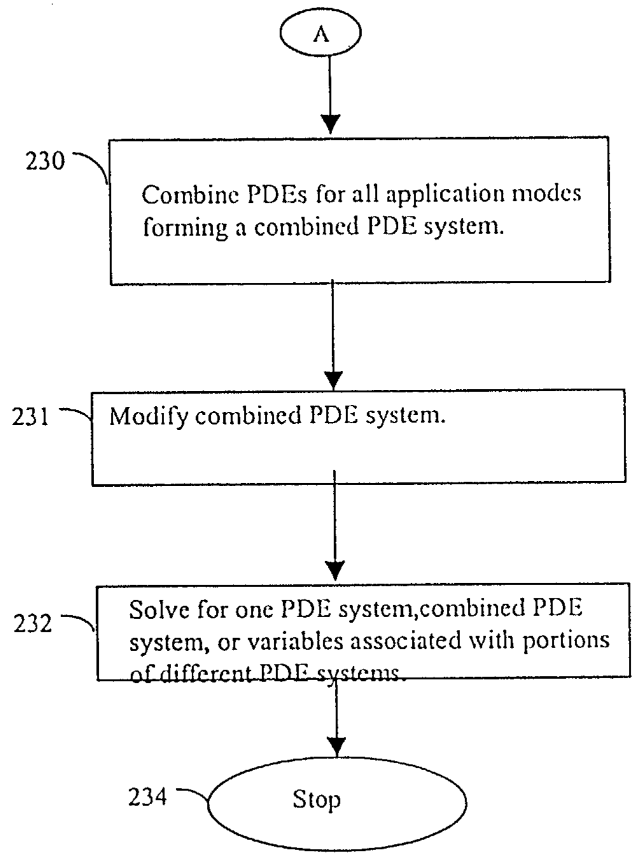
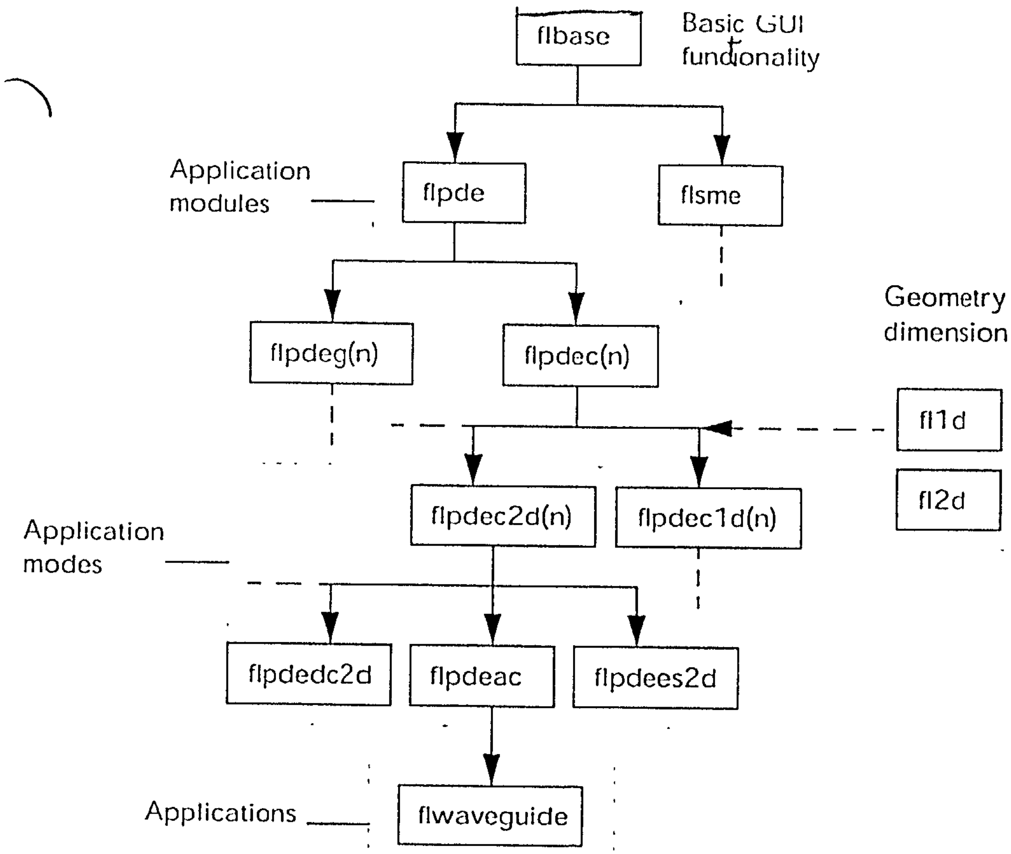

A
230
Combine PDEs for all application modes forming a combined PDE system.
231
Modify combined PDE system.
232
Solve for one PDE system,combined PDE system, or variables associated with portions of different PDE systems.
234
Stop

FIGURE 23

500



The class hierarchy of FEMLAB

FIGURE 24

1-D Physics Application Modes

Application mode	Class name	Parent class
Diffusion	`flpdedf1d`	`flpdedf`
Heat Transfer	`flpdeht1d`	`flpdeht`

502

1-D PDE Application Modes

Application mode	Class name	Parent class
Coefficient PDE model, n variables	`flpdec1d`*(n)*	`flpdec`*(n)*
General PDE model, n variables	`flpdeg1d`*(n)*	`flpdeg`*(n)*

504

FIGURE 25

2-D Physics Application Modes

Application mode	Class name	Parent class
AC Power Electromagnetics	flpdeac	flpdec2d
Conductive Media DC	flpdedc2d	flpdedc
Diffusion	flpdedf2d	flpdedf
Electrostatics	flpdees2d	flpdees
Magnetostatics	flpdems2d	flpdems
Heat Transfer	flpdeht2d	flpdeht
Incompressible Navier-Stokes	flpdens2d	flpdens
Structural Mechanics, Plane Stress	flpdeps	flpdec2d
Structural Mechanics, Plane Strain	flpdepn	flpdec2d

506

PDE Application Modes

Application mode	Class name	Parent class
Coefficient PDE model, n variables	flpdec2d *(n)*	flpdec *(n)*
General PDE model, n variables	flpdeg2d *(n)*	flpdeg *(n)*

508

FIGURE 26

Application Object Properties

Property name	Description	Data type
dim	Names of the dependent variables	Cell array of strings
form	PDE form	String (coefficient/general)
name	Application name	String
parent	Parent class names	String, cell array of strings, or the empty matrix
sdim	Names of the independent variables (space dimensions)	Cell array of strings
submode	Name of current submode	String (std/wave)
tdiff	Time differentiation flag	String (on/off)

51

FIGURE 27

512

```
function obj = myapp()
%MYAPP Constructor for a FEMLAB application object.

obj.name = 'My first FEMLAB application';
obj.parent = 'flpdeht2d';

% MYAPP is a subclass of FLPDEHT2D:
p1 = flpdeht2d;
obj = class(obj,'myapp',p1);
set(obj,'dim',default_dim(obj));
```

FIGURE 28

514

Physics Modeling Methods

Function	Purpose
appspec	Return application specifications.
bnd_compute	Convert application-dependent boundary conditions to generic boundary coefficients.
default_bnd	Default boundary conditions.
default_dim	Default names of dependent variables.
default_equ	Default PDE coefficients/Material parameters.
default_init	Default initial conditions.
default_sdim	Default space dimension variables.
default_var	Default application scalar variables.
dim_compute	Return dependent variables for an application.
equ_compute	Convert application-dependent material parameters to generic PDE coefficients.
form_compute	Return PDE form.
init_compute	Convert application-dependent initial conditions to generic initial conditions.
posttable	Define assigned variable names and post-processing information.

FIGURE 29

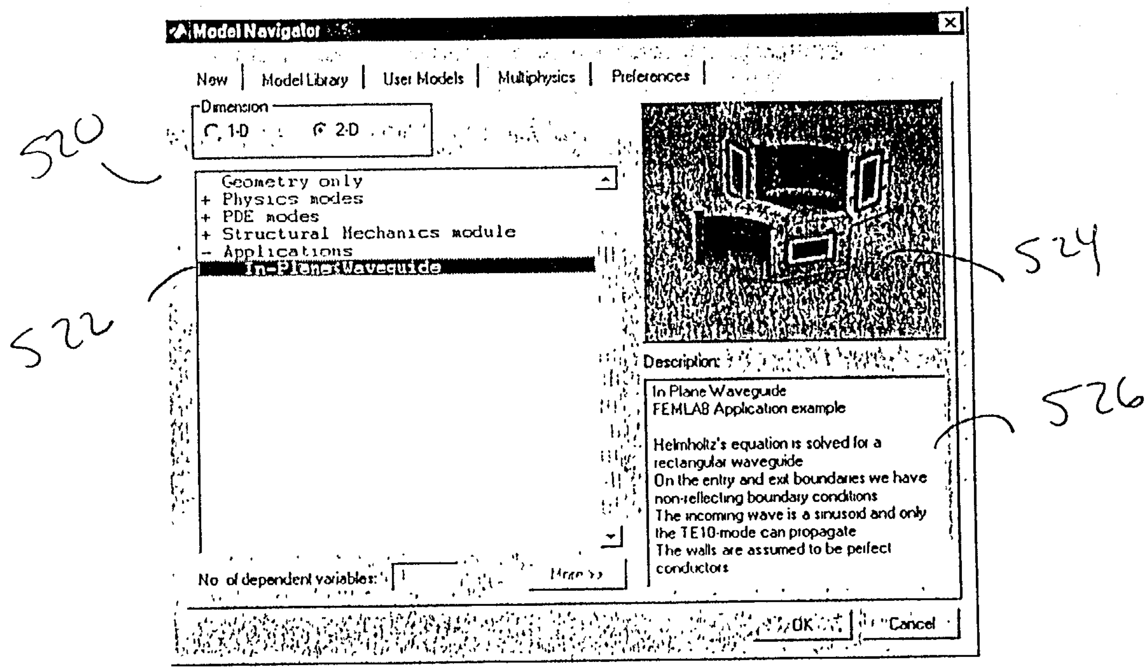

Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Dimension
1-D
2-D
Geometry only
+ Physics modes
+ PDE modes
+ Structural Mechanics module
- Applications
In-Plane Waveguide
Description:
In Plane Waveguide
FEMLAB Application example
Helmholtz's equation is solved for a rectangular waveguide
On the entry and exit boundaries we have non-reflecting boundary conditions
The incoming wave is a sinusoid and only the TE10-mode can propagate
The walls are assumed to be perfect conductors
No. of dependent variables:
OK
Cancel
520
522
524
526

FIGURE 30

530 $$\Delta E_z + (2\pi i k)^2 E_z = 0$$

532 $$k = \frac{1}{\lambda} = \frac{f}{c}$$

534 $$\bar{n} \cdot (\nabla E_z) + 2\pi i k_x E_z = 4\pi i k_x \sin\left(\frac{\pi}{d}(y - y_0)\right)$$

536 $$k^2 = k_x^2 + k_y^2$$

538 $$k_x = \sqrt{\frac{1}{\lambda^2} - \frac{1}{(2d)^2}}$$

540 $$n \cdot (\nabla E_z) + 2\pi i k_x E_z = 0$$

542 $$E_z = 0$$

544 $$f_c = \frac{c}{2d}$$

FIGURE 31

550

```
function obj = flwaveguide(varargin)
%FLWAVEGUIDE Constructor for a Waveguide application object.

obj.name = 'In-Plane Waveguide';
obj.parent = 'flpdeac';

% FLWAVEGUIDE is a subclass of FLPDEAC:
p1 = flpdeac;
obj = class(obj,'flwaveguide',p1);
set(obj,'dim',default_dim(obj));
```

FIGURE 32

552

fem.user fields

Field	Description
`geomparam`	1-by-2 structure of geometry parameters.
`entrybnd`	Index to the entry boundary.
`exitbnd`	Index to the exit boundary.
`freqs`	Frequency vector

FIGURE 33

554

fem.user fields

Field	Description
`startpt`	Index of the lower left corner point of the waveguide.
`type`	Type of waveguide. (*'straight* or *'elbow*)

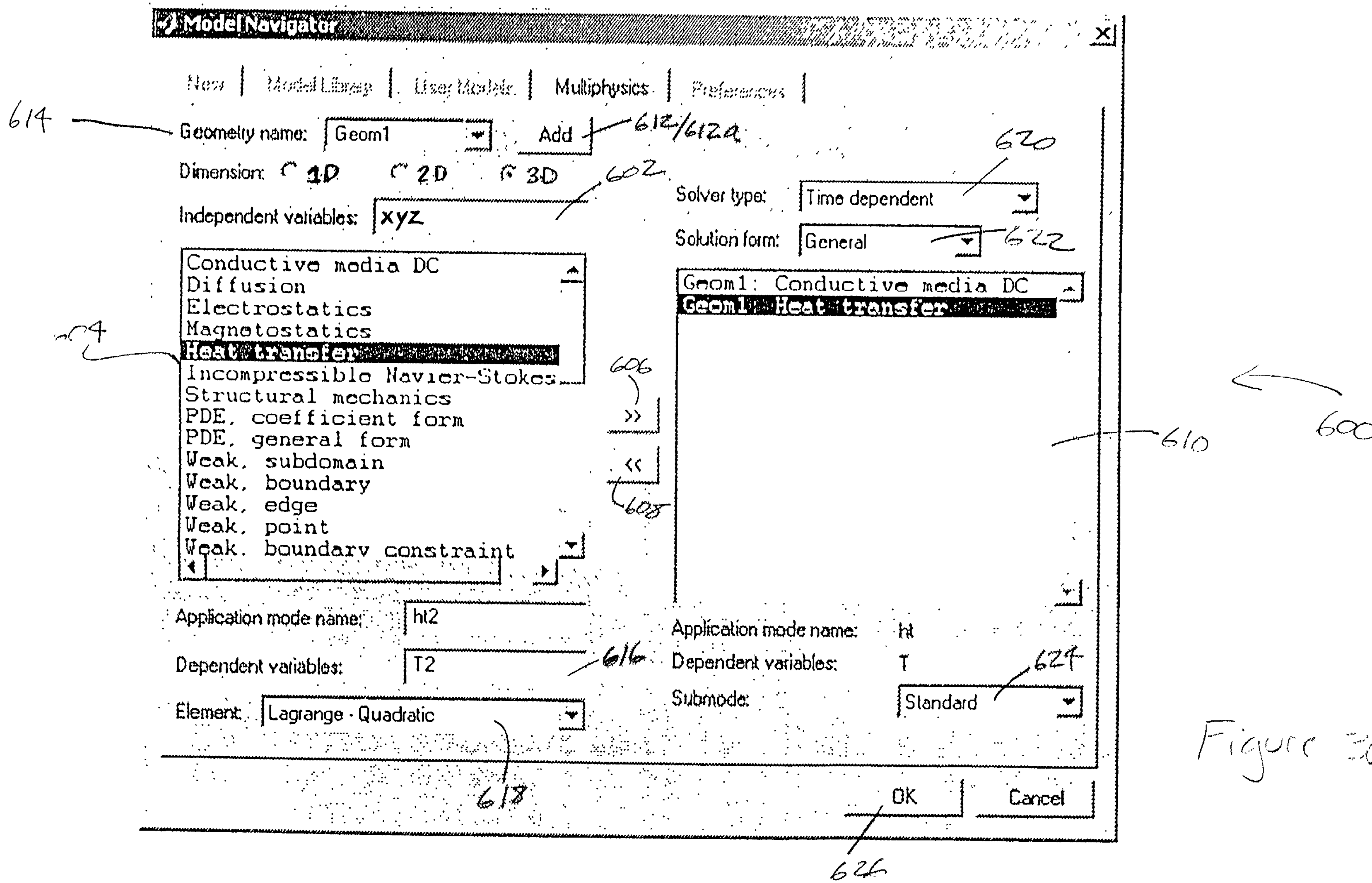
FIGURE 34

556

geomparam fields

Field	Description	Defaults for elbow	Defaults for straight
`entrylength`	Length of the entrance part of the waveguide.	0.1	0.1
`exitlength`	Length of the exit part of the waveguide.	0.1	Not used
`radius`	Outer radius of the waveguide bend.	0.05	Not used
`width`	Width of the waveguide.	0.025	0.025
`cavityflag`	Turn resonance cavity *on* or *off*.	0	0
`cavitywidth`	Width of the resonance cavity.	0.025	0.025
`postwidth`	Width of the protruding posts.	0.005	0.005
`postdepth`	Depth of the protruding posts.	0.005	0.005

FIGURE 35



Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Geometry name:
Geom1
Add
Dimension:
1D
2D
3D
Independent variables:
xyz
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Structural mechanics
PDE, coefficient form
PDE, general form
Weak, subdomain
Weak, boundary
Weak, edge
Weak, point
Weak, boundary constraint
>>
<<
Solver type:
Time dependent
Solution form:
General
Geom1: Conductive media DC
Geom1: Heat transfer
Application mode name:
ht2
Dependent variables:
T2
Element:
Lagrange - Quadratic
Application mode name:
ht
Dependent variables:
T
Submode:
Standard
OK
Cancel
614
612/612a
602
604
606
608
610
600
620
622
616
618
624
626

Figure 36

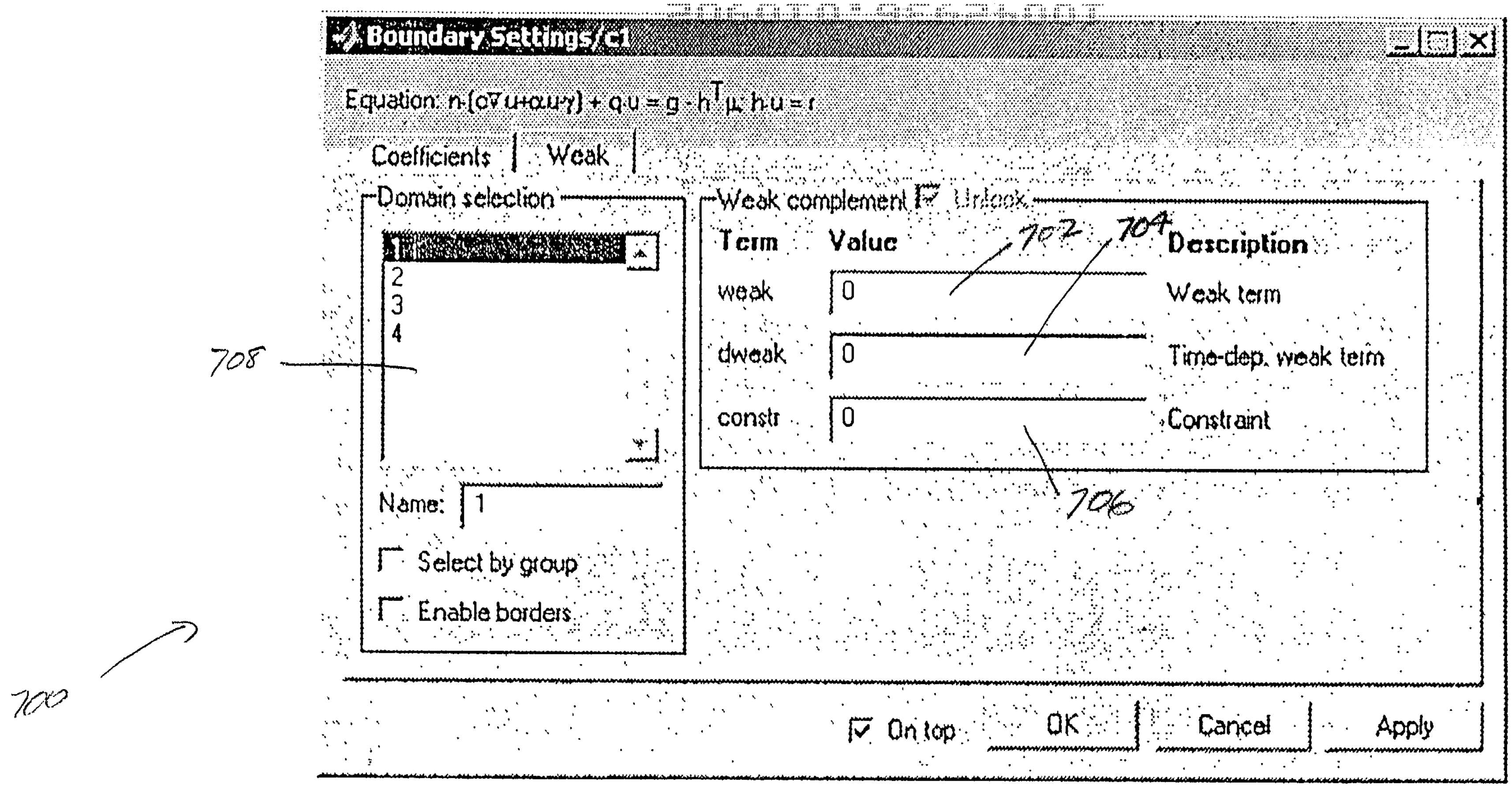

Boundary Settings/c1
Equation: n·(c∇u+αu-γ) + q·u = g - h^Tμ; h·u = r
Coefficients
Weak
Domain selection
1
2
3
4
Name: 1
Select by group
Enable borders
Weak complement
Unlock
Term
Value
Description
weak
0
Weak term
dweak
0
Time-dep. weak term
constr
0
Constraint
On top
OK
Cancel
Apply
702
704
706
708
700

Figure 37

800

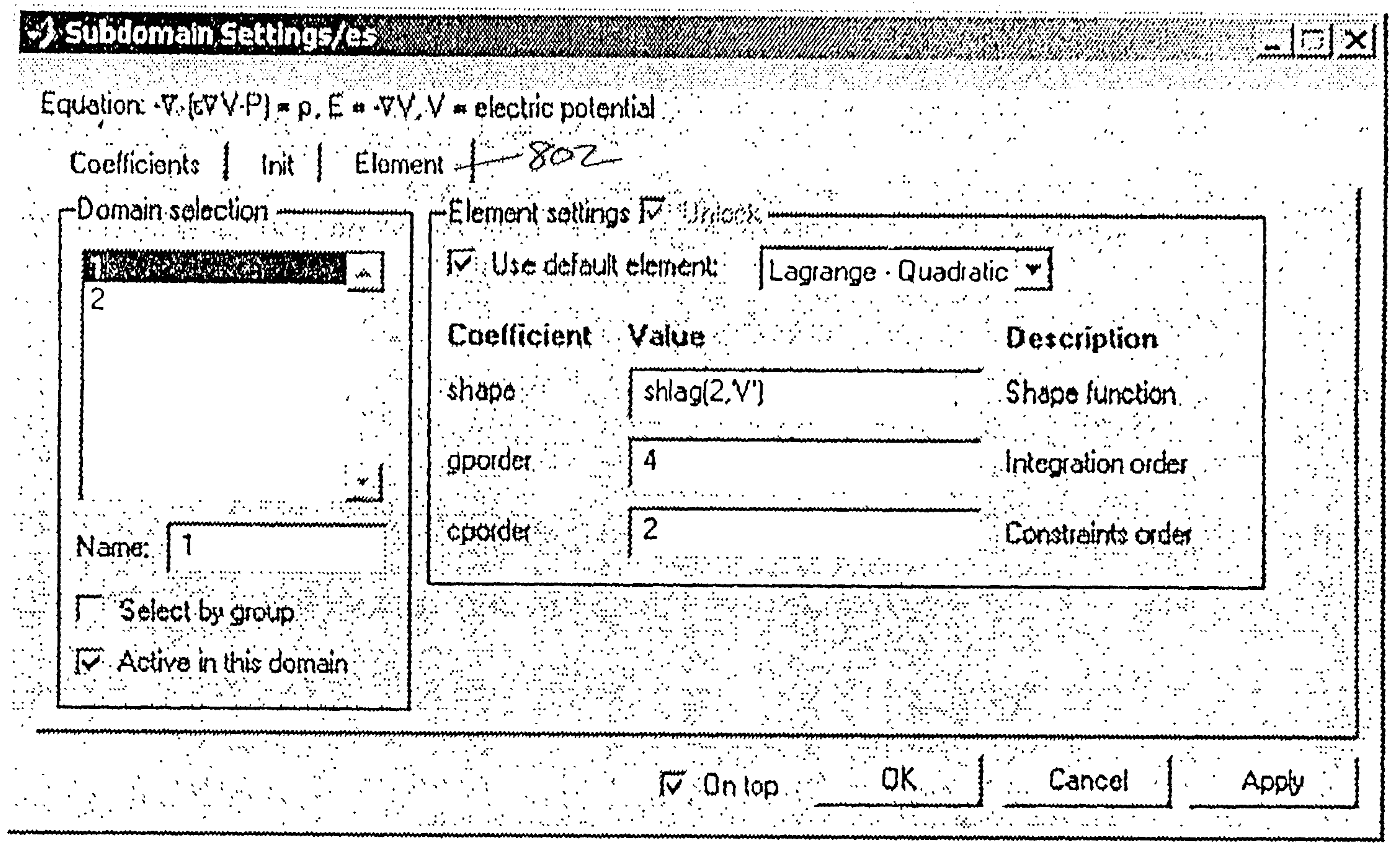
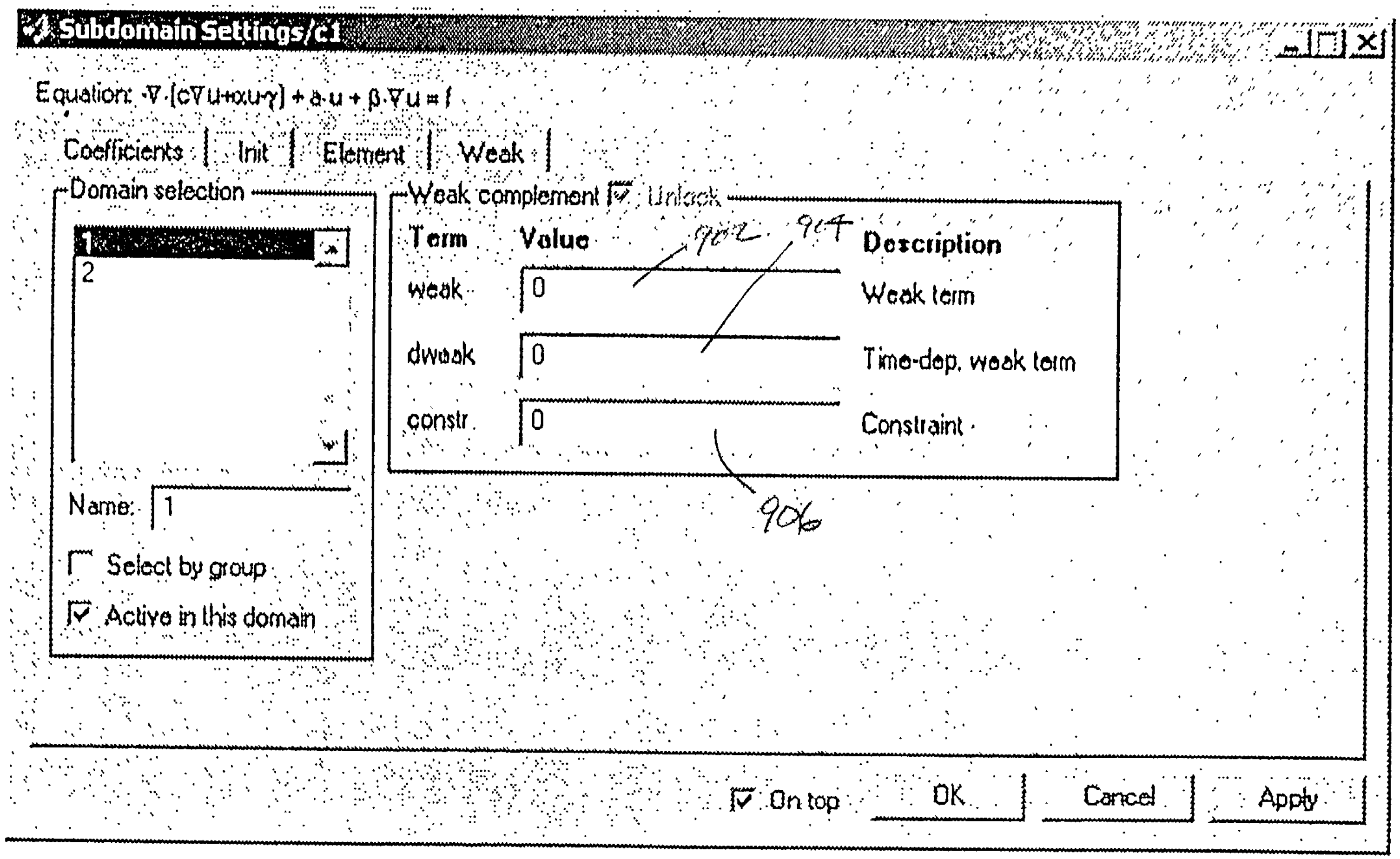

Subdomain Settings/es
Equation: -∇·(ε∇V-P) = ρ, E = -∇V, V = electric potential
Coefficients
Init
Element
802
Domain selection
1
2
Name: 1
Select by group
Active in this domain
Element settings
Unlock
Use default element:
Lagrange - Quadratic
Coefficient
Value
Description
shape
shlag(2,'V')
Shape function
gporder
4
Integration order
cporder
2
Constraints order
On top
OK
Cancel
Apply

Figure 8



↑ 900

Figure 39

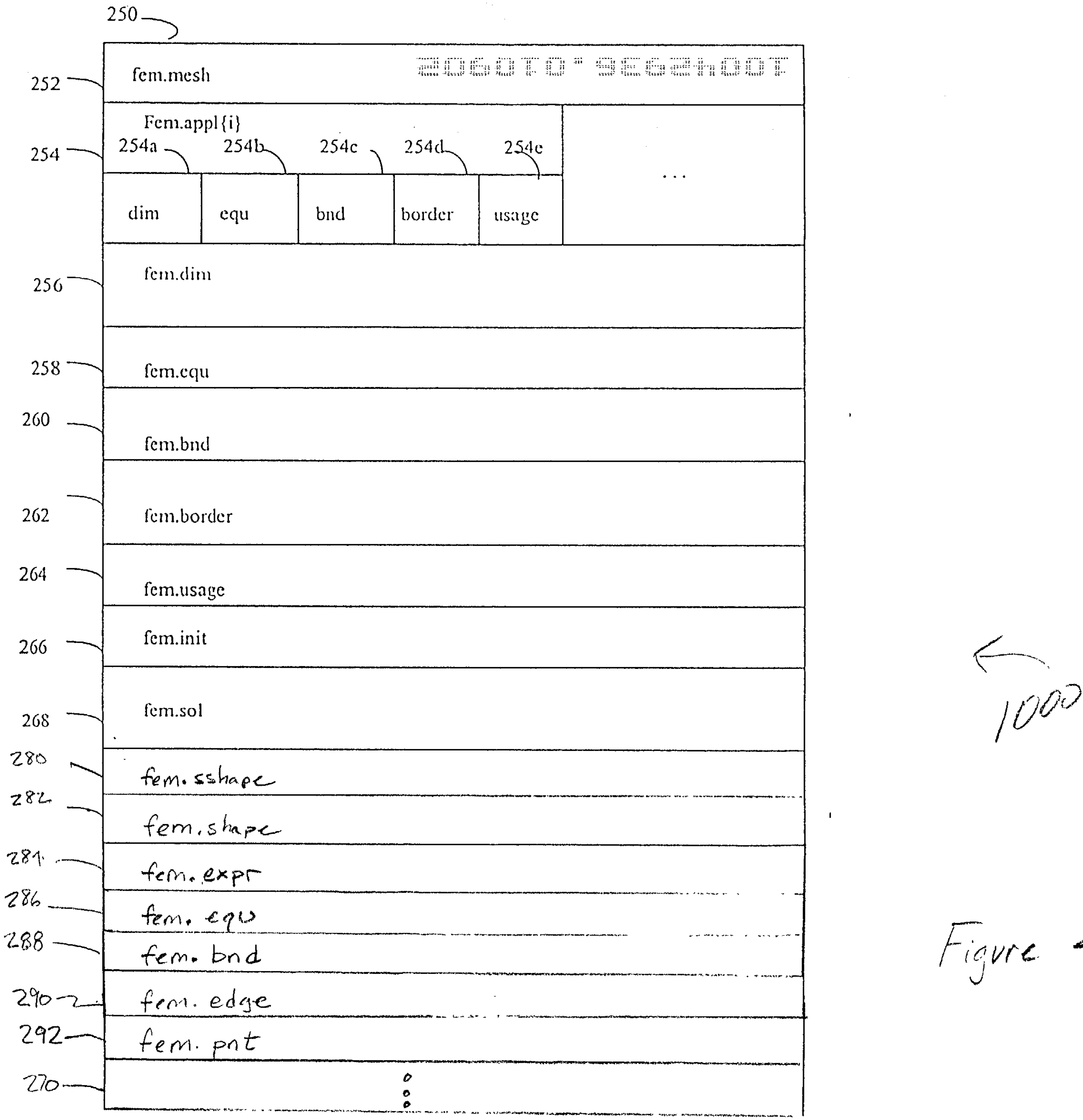

250
252
fem.mesh
254
Fem.appl{i}
254a
254b
254c
254d
254e
dim
equ
bnd
border
usage
...
256
fem.dim
258
fem.equ
260
fem.bnd
262
fem.border
264
fem.usage
266
fem.init
268
fem.sol
280
fem.sshape
282
fem.shape
284
fem.expr
286
fem.equ
288
fem.bnd
290
fem.edge
292
fem.pnt
270
1000

Figure 40

1102

$$
\left\{
\begin{aligned}
0 &= \int_{\Omega} W^{(2)}\,dA + \int_{B} W^{(1)}\,ds + \sum_{P} W^{(0)} + \\
&+ \int_{\Omega} v_l \frac{\partial R_m^{(2)}}{\partial u_l}\,\mu_m^{(2)}\,dA + \int_{B} v_l \frac{\partial R_m^{(1)}}{\partial u_l}\,\mu_m^{(1)}\,ds + \sum_{P} v_l \frac{\partial R_m^{(0)}}{\partial u_l}\,\mu_m^{(0)}
\end{aligned}
\right.
$$

1104

$$
\left\{
\begin{aligned}
0 &= R^{(2)} \quad \text{on } \Omega \\
0 &= R^{(1)} \quad \text{on B} \\
0 &= R^{(0)} \quad \text{on P}
\end{aligned}
\right.
$$

1100

Figure 41

$$W_l^{(n)} = W_l^{(n)} + \Gamma_{lj}\frac{\partial v_l}{\partial x_j} + F_l v_l$$

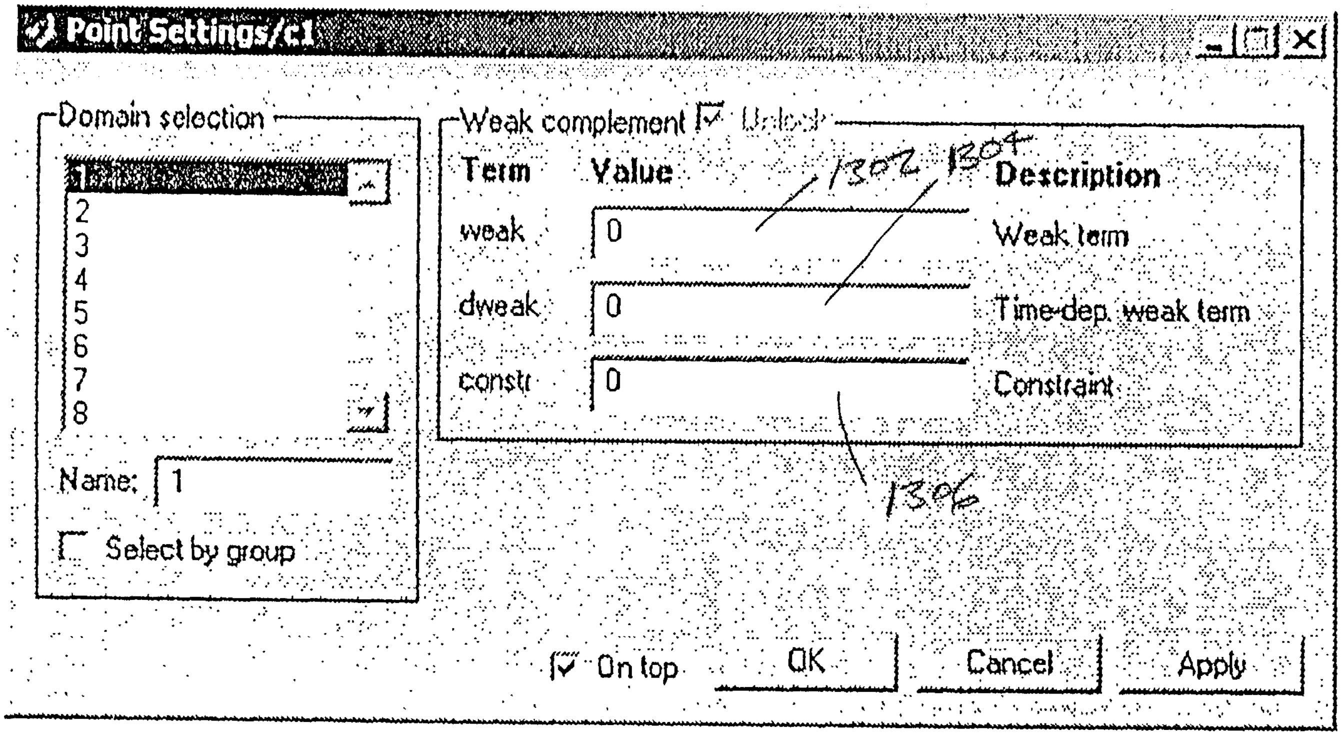
$$W_l^{(nt)} = W_l^{(nt)} + d_{alk}\frac{\partial u_k}{\partial t} v_l$$

$$W_l^{(n-1)} = W_l^{(n-1)} + G_l v_l$$

$$R_m^{(n)} = R_m$$

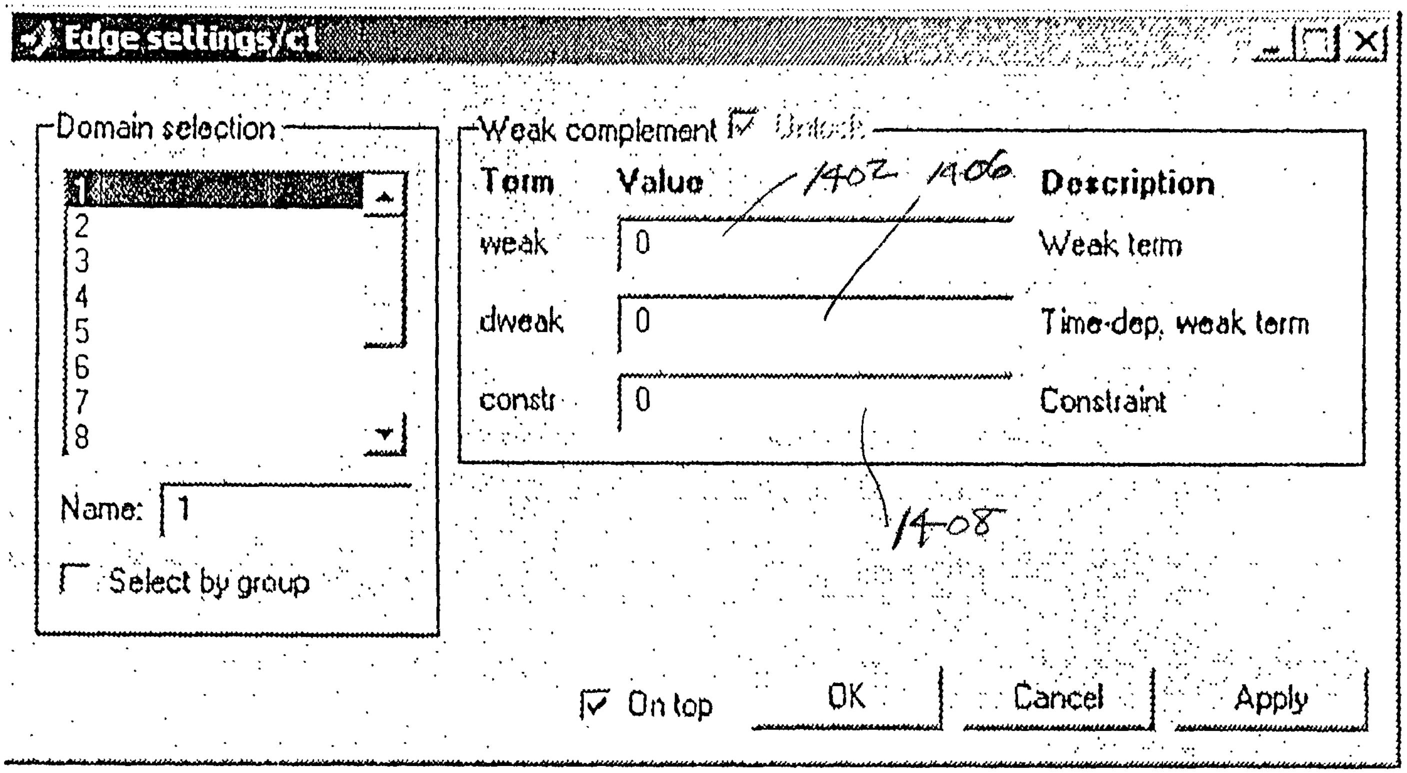
1200

Figure 42



1300

Figure 4C



1400

Figure 14

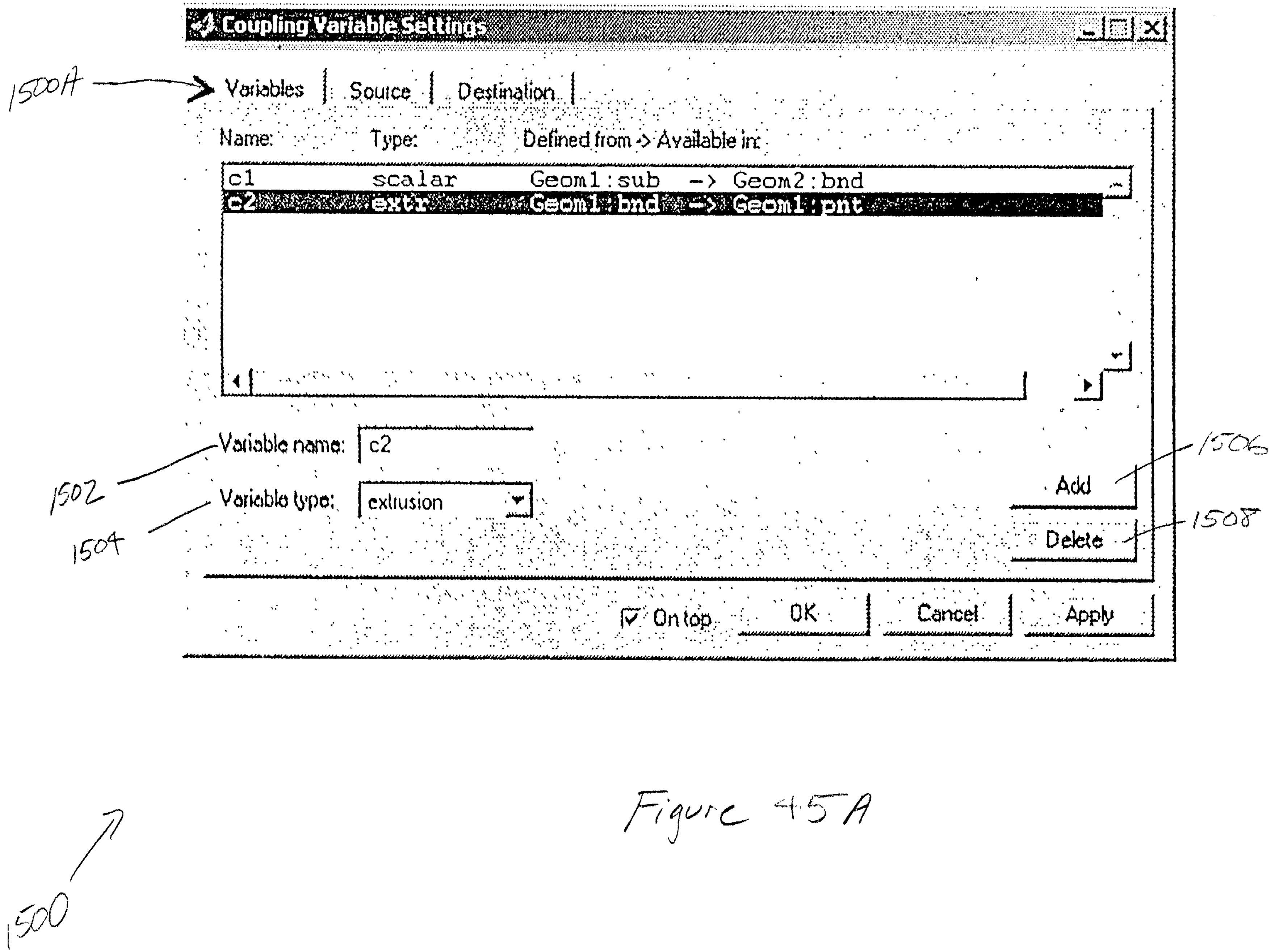

1500A
Coupling Variable Settings
Variables
Source
Destination
Name:
Type:
Defined from -> Available in:
c1 scalar Geom1:sub -> Geom2:bnd
c2 extr Geom1:bnd -> Geom1:pnt
Variable name:
c2
Variable type:
extrusion
Add
Delete
On top
OK
Cancel
Apply
1502
1504
1506
1508
1500

Figure 45A

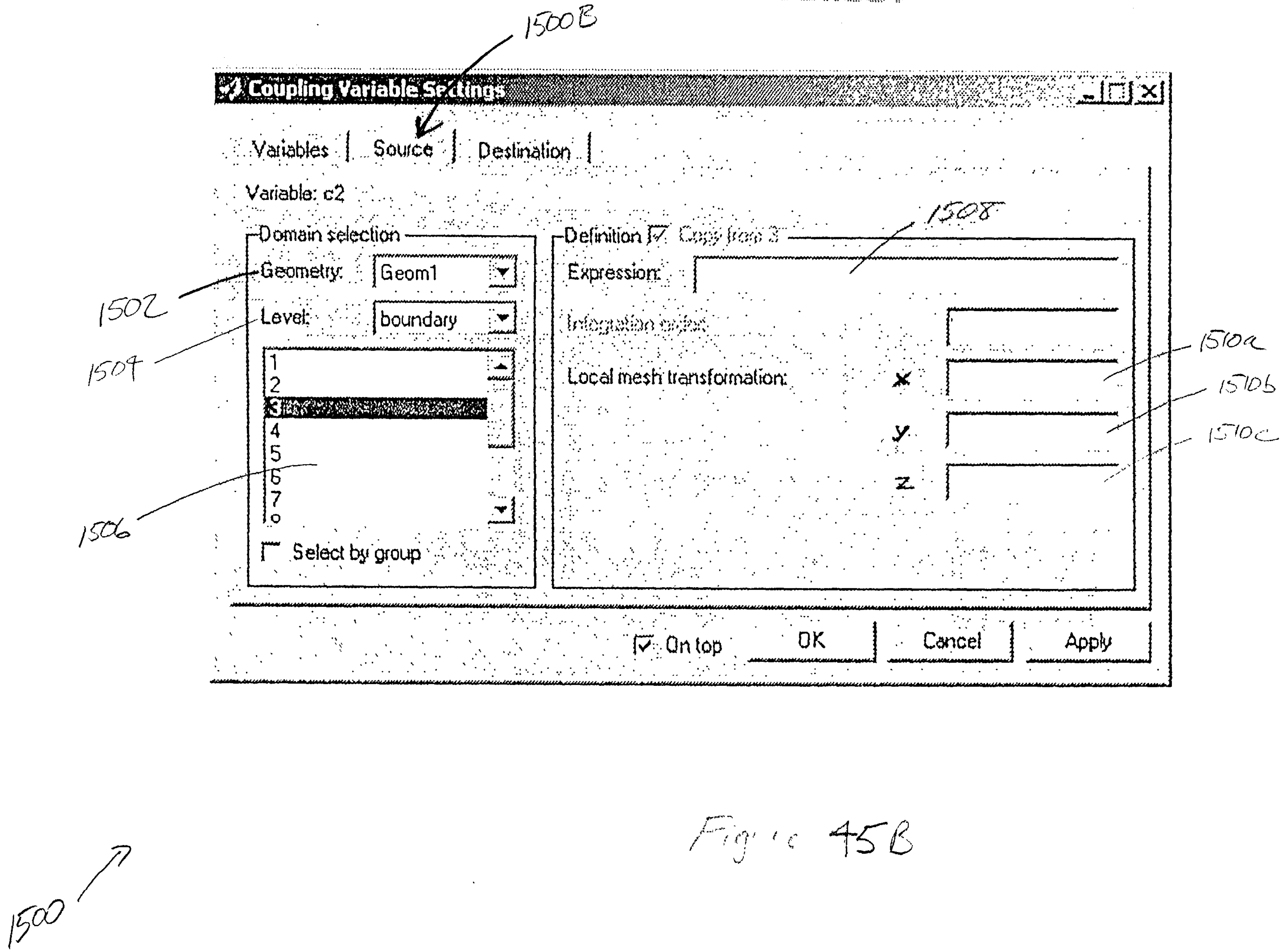
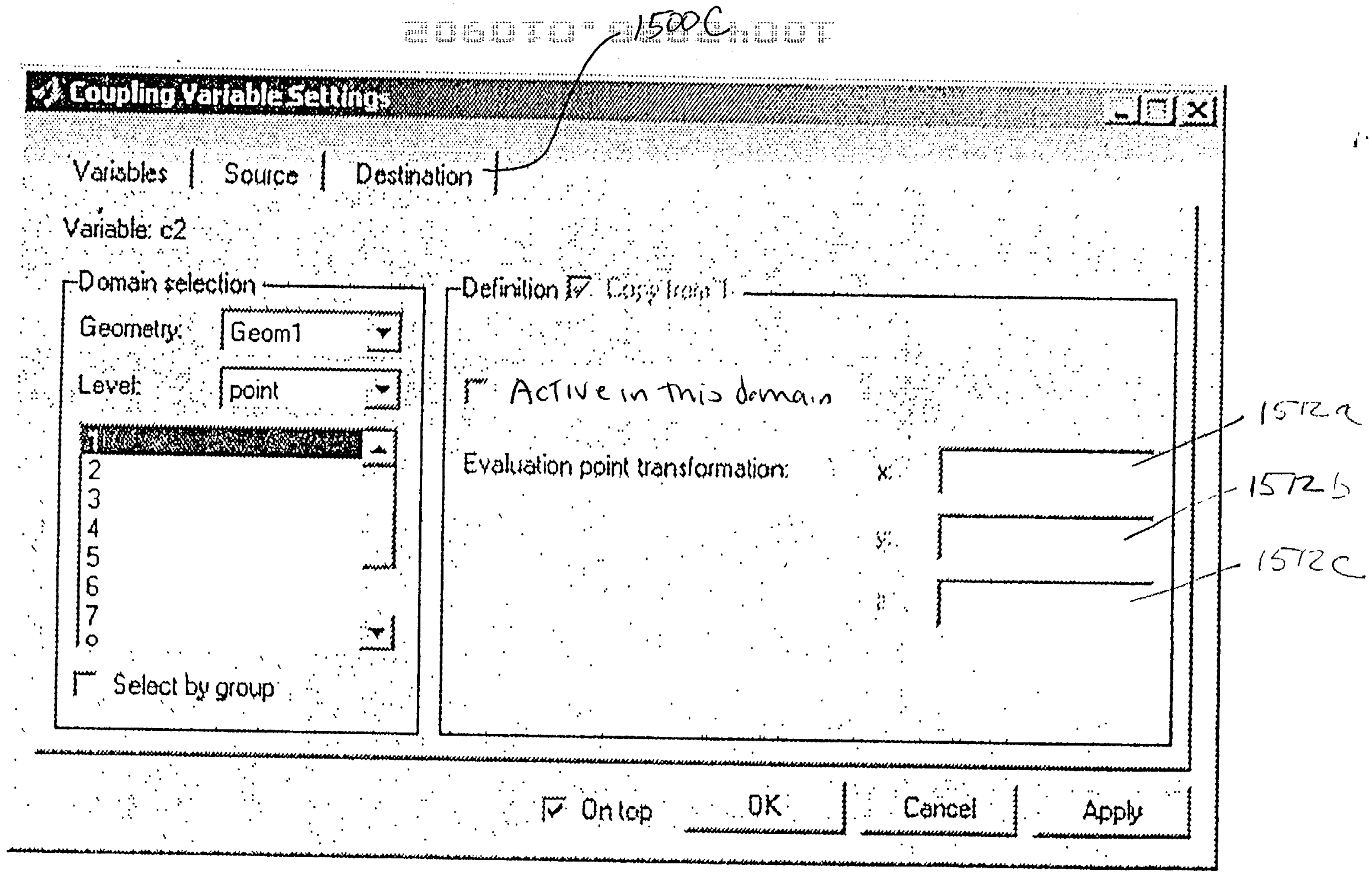

1500B
Coupling Variable Settings
Variables
Source
Destination
Variable: c2
Domain selection
Geometry:
Geom1
Level:
boundary
1
2
3
4
5
6
7
Select by group
Definition
Expression:
Local mesh transformation:
x
y
z
On top
OK
Cancel
Apply
1502
1504
1506
1508
1510a
1510b
1510c
1500

Figure 45B



1500

Figure 45C

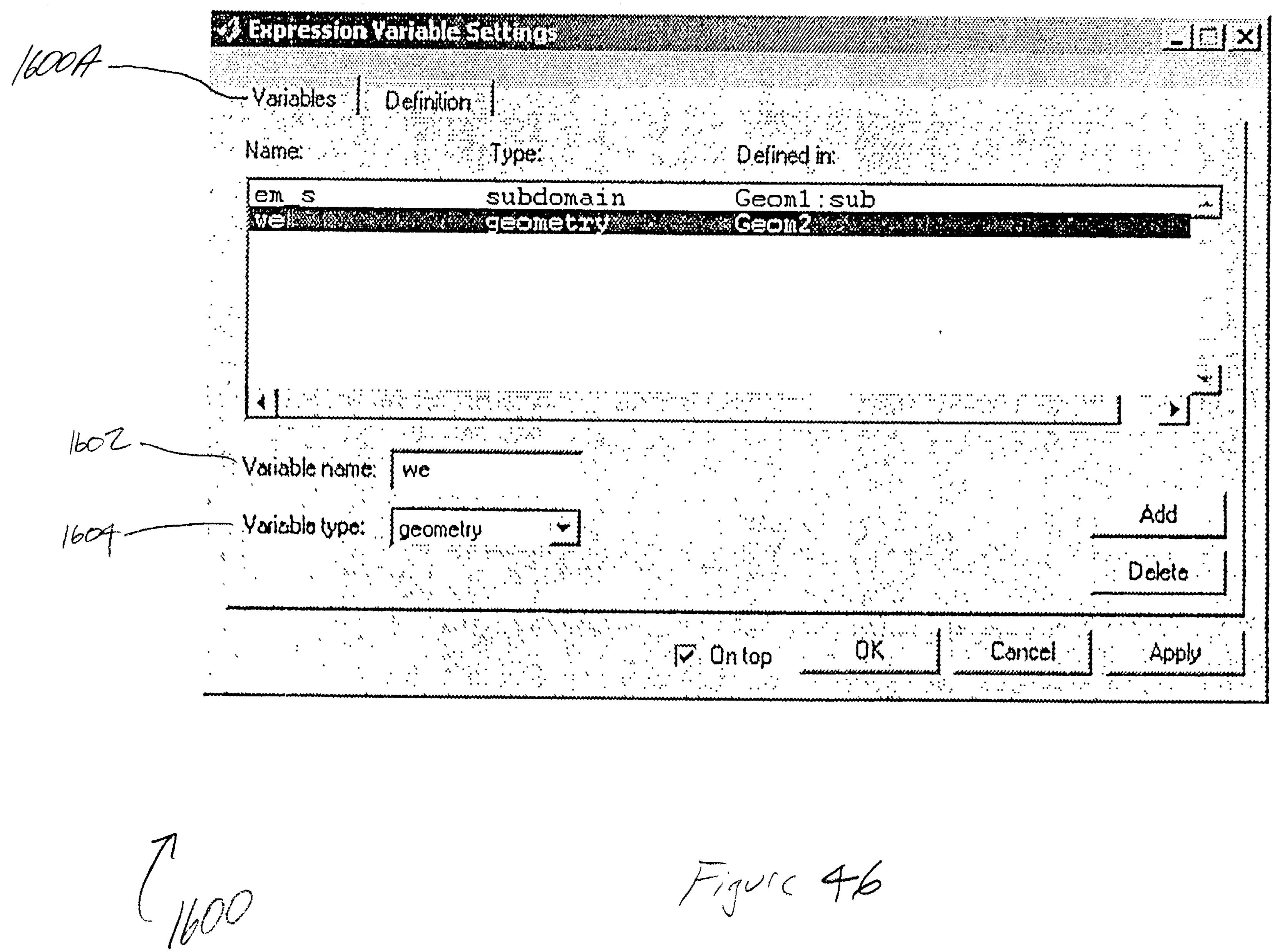

Expression Variable Settings
Variables
Definition
Name:
Type:
Defined in:
em_s
subdomain
Geom1:sub
we
geometry
Geom2
Variable name:
we
Variable type:
geometry
Add
Delete
On top
OK
Cancel
Apply
1600A
1602
1604
1600

Figure 46

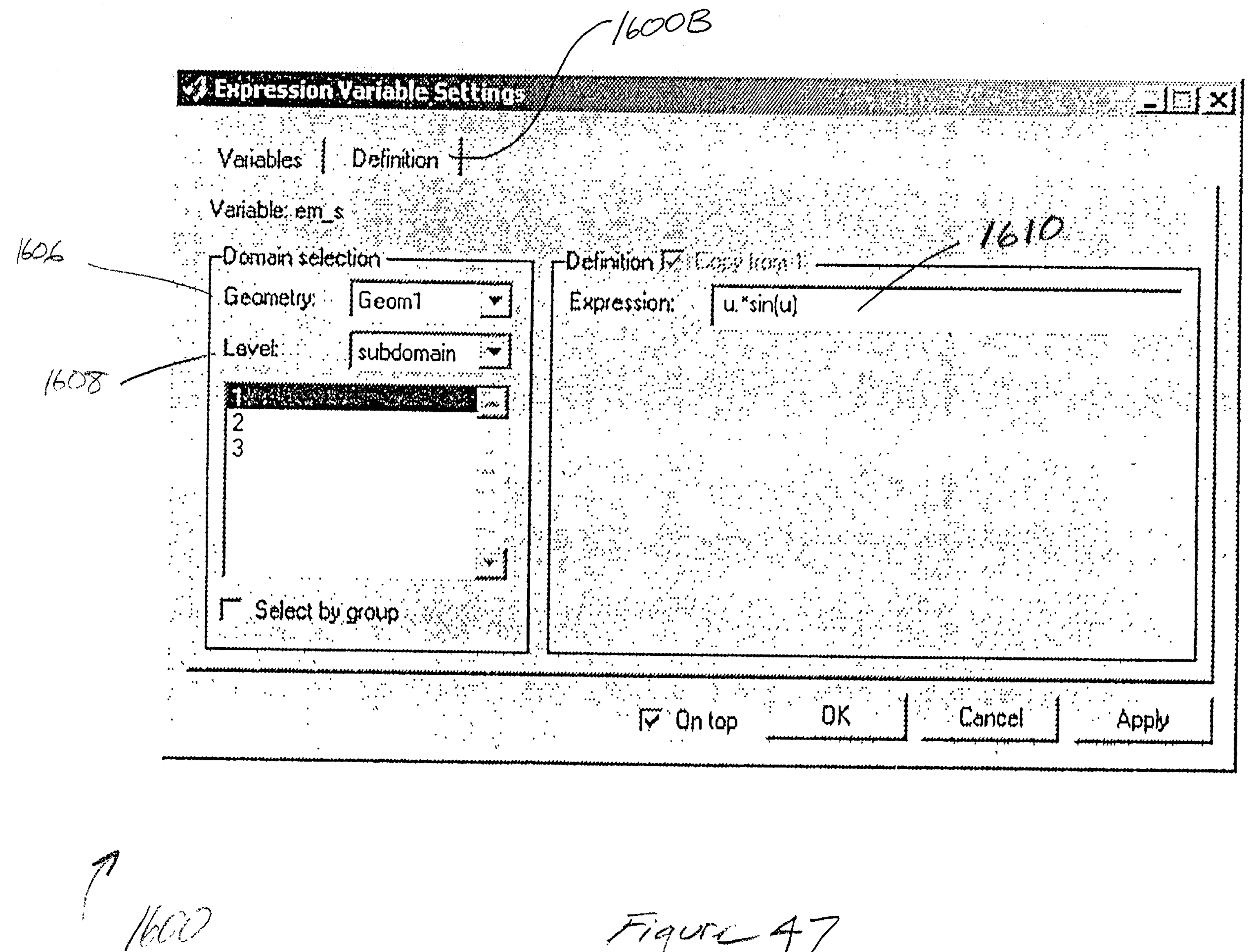

1600B
Expression Variable Settings
Variables
Definition
Variable: em_s
1606
1608
Domain selection
Geometry:
Geom1
Level:
subdomain
1
2
3
Select by group
1610
Definition
Expression:
u.*sin(u)
On top
OK
Cancel
Apply
1600

Figure 47

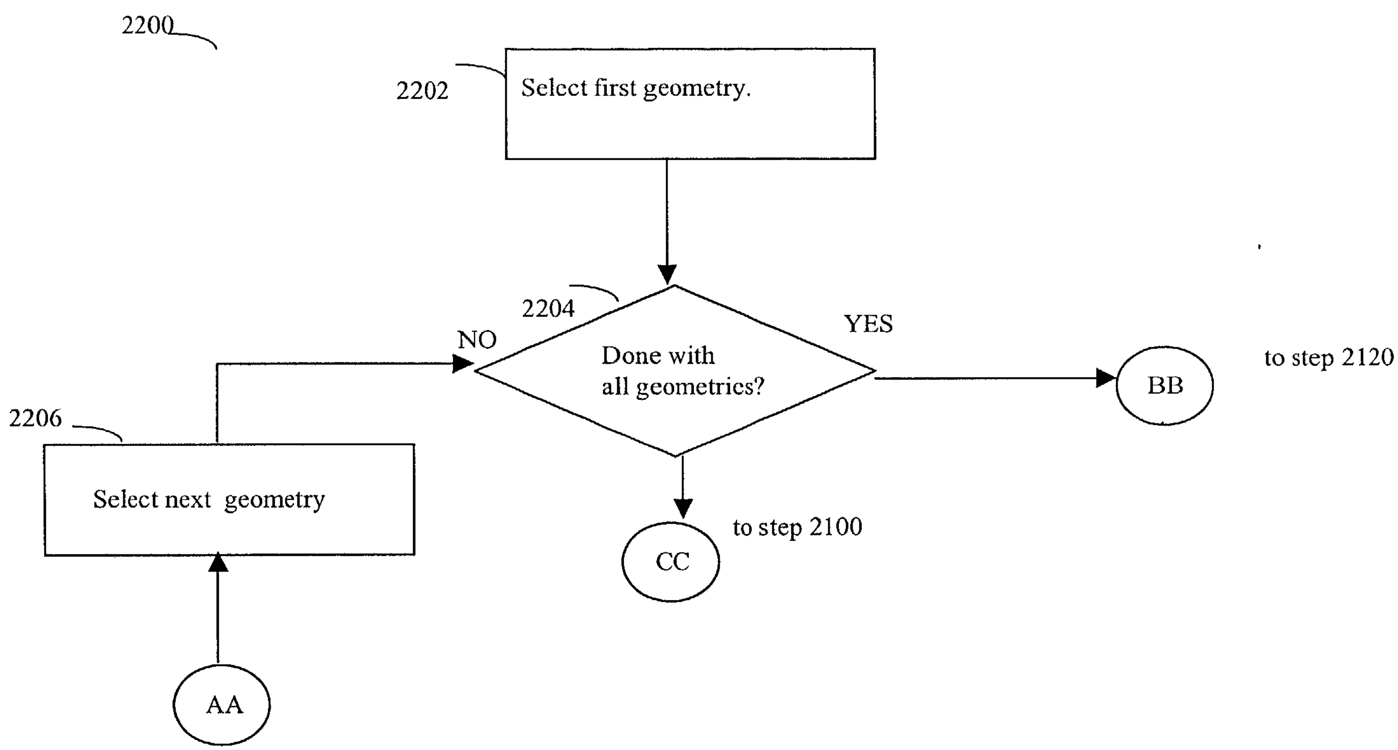

2200
2202
Selcct first geometry.
2204
Done with
all geometrics?
YES
NO
BB
to step 2120
CC
to step 2100
2206
Select next geometry
AA

FIGURE 48

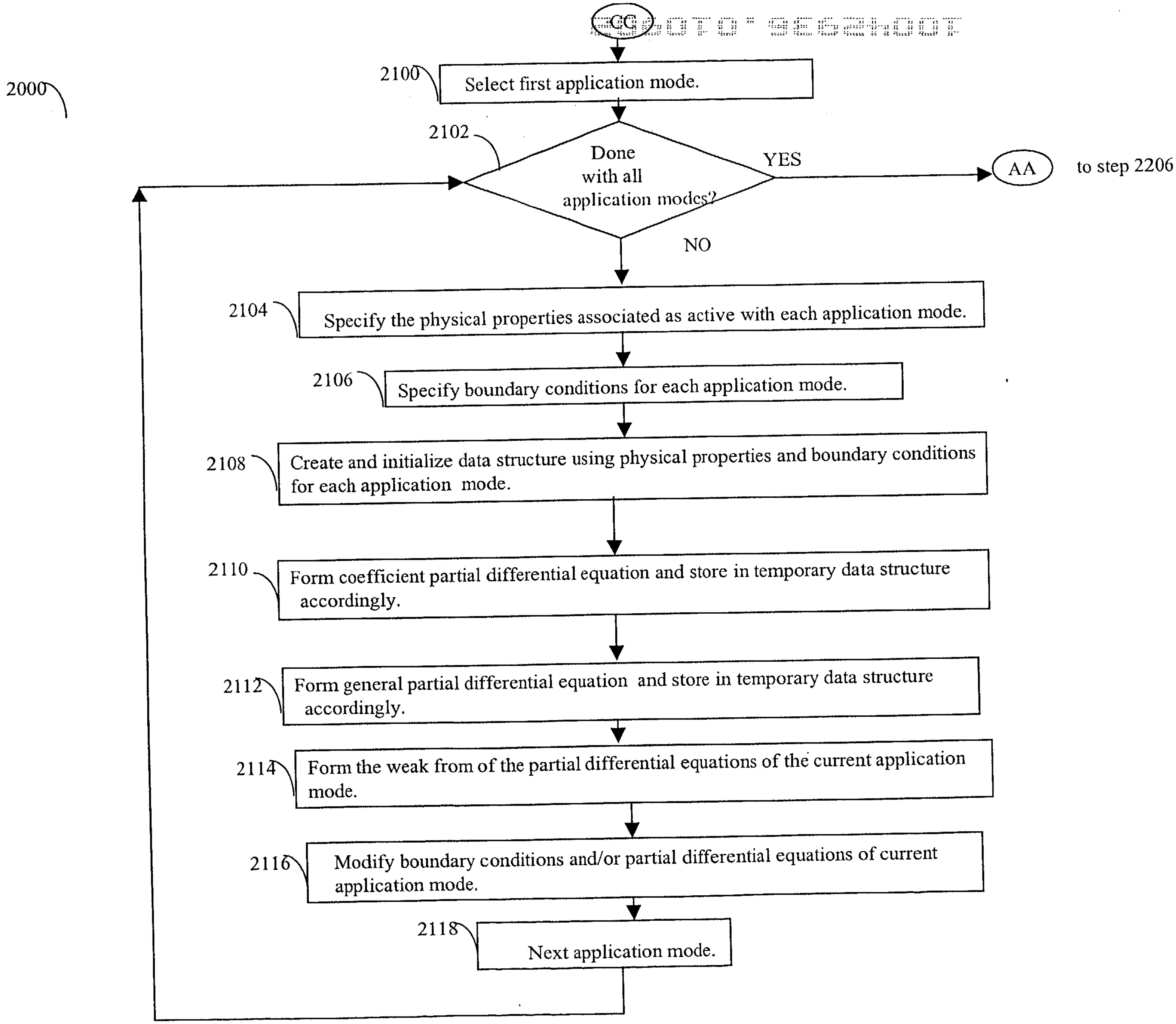

2000
CC
2100
Select first application mode.
2102
Done with all application modes?
YES
AA
to step 2206
NO
2104
Specify the physical properties associated as active with each application mode.
2106
Specify boundary conditions for each application mode.
2108
Create and initialize data structure using physical properties and boundary conditions for each application mode.
2110
Form coefficient partial differential equation and store in temporary data structure accordingly.
2112
Form general partial differential equation and store in temporary data structure accordingly.
2114
Form the weak from of the partial differential equations of the current application mode.
2116
Modify boundary conditions and/or partial differential equations of current application mode.
2118
Next application mode.

FIGURE 49

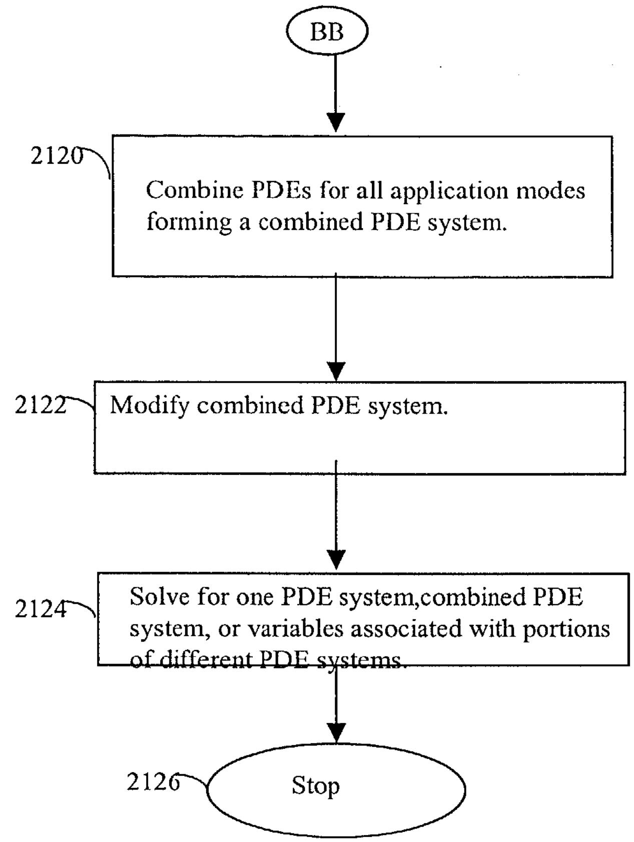

BB
2120
Combine PDEs for all application modes forming a combined PDE system.
2122
Modify combined PDE system.
2124
Solve for one PDE system,combined PDE system, or variables associated with portions of different PDE systems.
2126
Stop

FIGURE 50

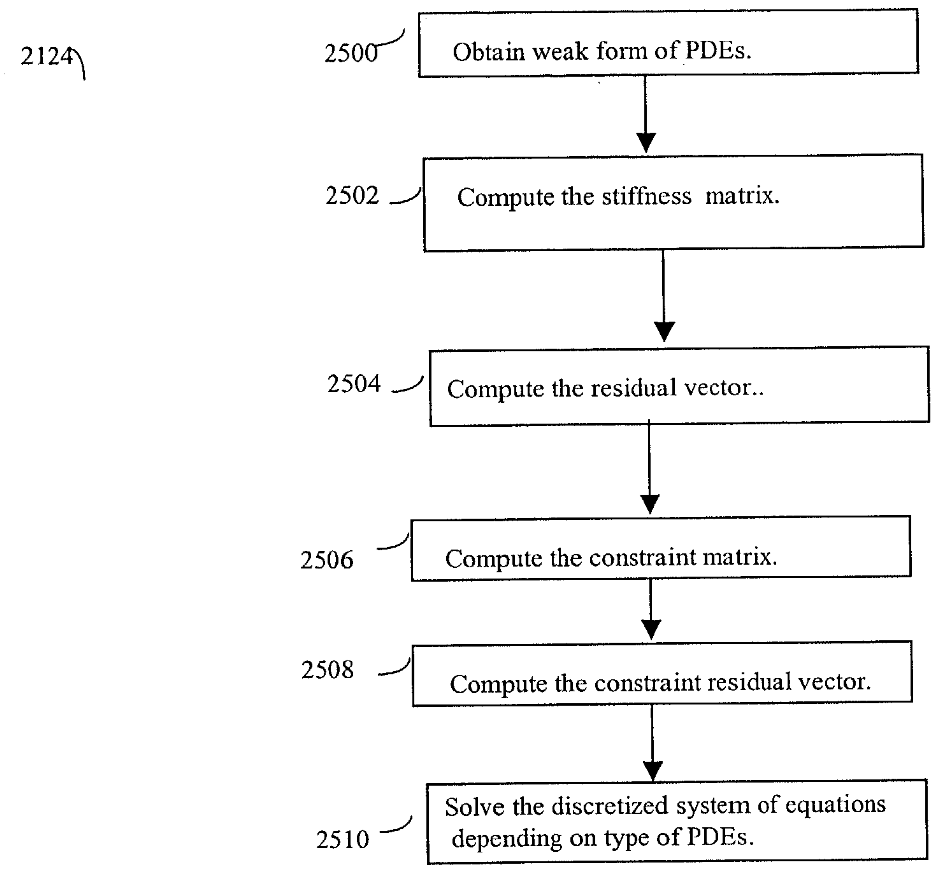
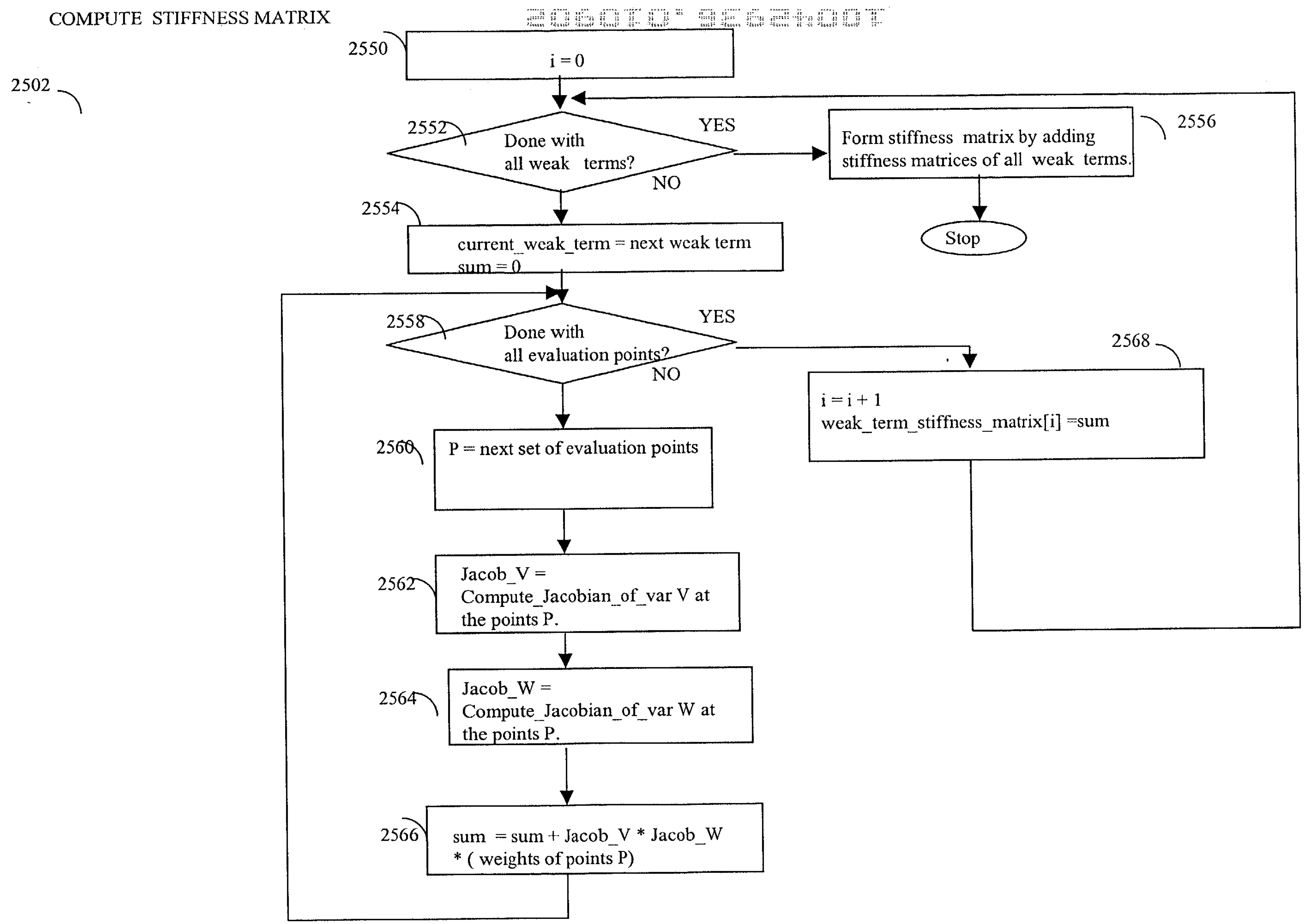

2124
2500
Obtain weak form of PDEs.
2502
Compute the stiffness matrix.
2504
Compute the residual vector..
2506
Compute the constraint matrix.
2508
Compute the constraint residual vector.
2510
Solve the discretized system of equations depending on type of PDEs.

FIGURE 51



COMPUTE STIFFNESS MATRIX
2502
2550
i = 0
2552
Done with all weak terms?
YES
NO
2556
Form stiffness matrix by adding stiffness matrices of all weak terms.
Stop
2554
current_weak_term = next weak term
sum = 0
2558
Done with all evaluation points?
YES
NO
2568
i = i + 1
weak_term_stiffness_matrix[i] =sum
2560
P = next set of evaluation points
2562
Jacob_V = Compute_Jacobian_of_var V at the points P.
2564
Jacob_W = Compute_Jacobian_of_var W at the points P.
2566
sum = sum + Jacob_V * Jacob_W * (weights of points P)

FIGURE 52

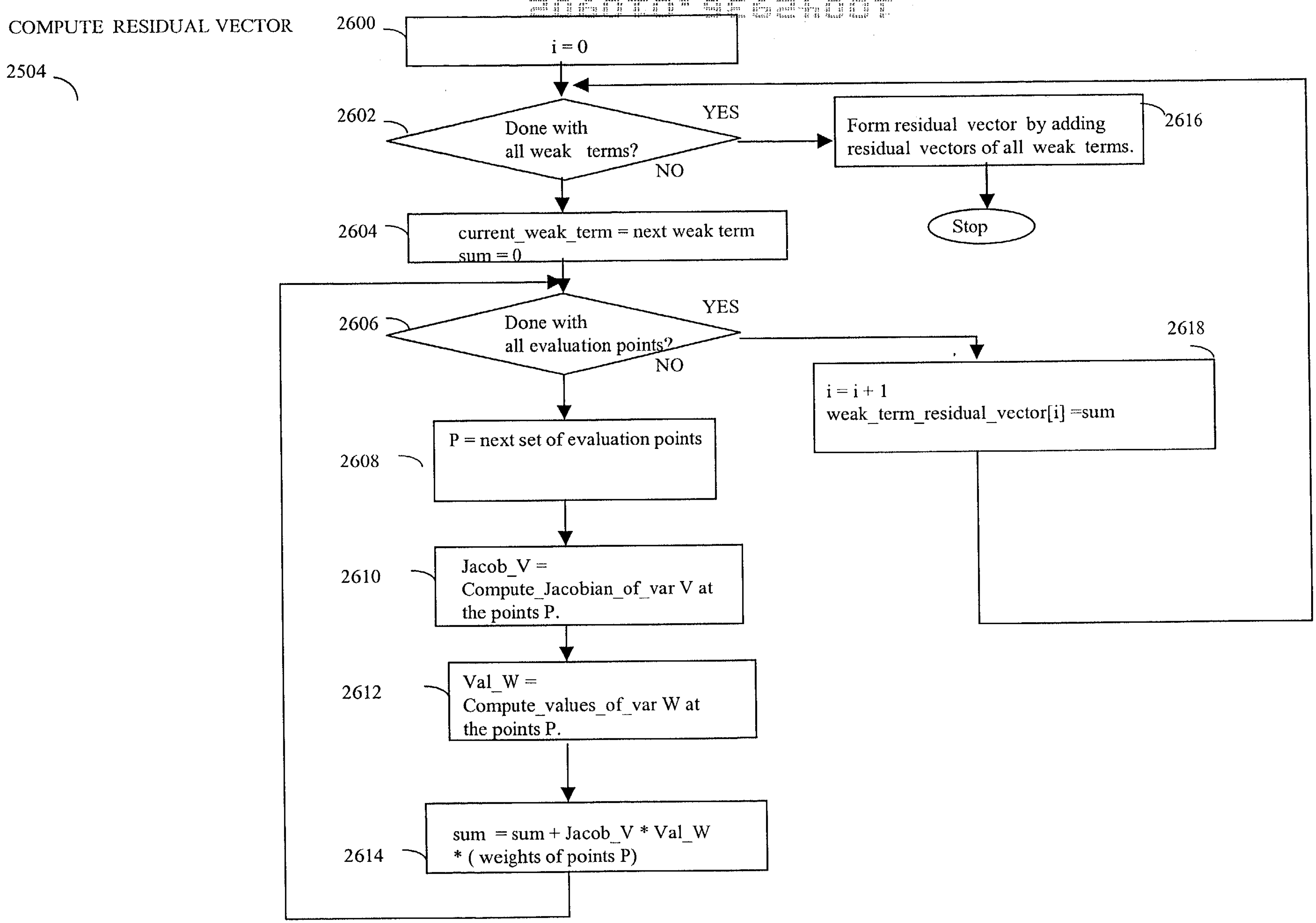
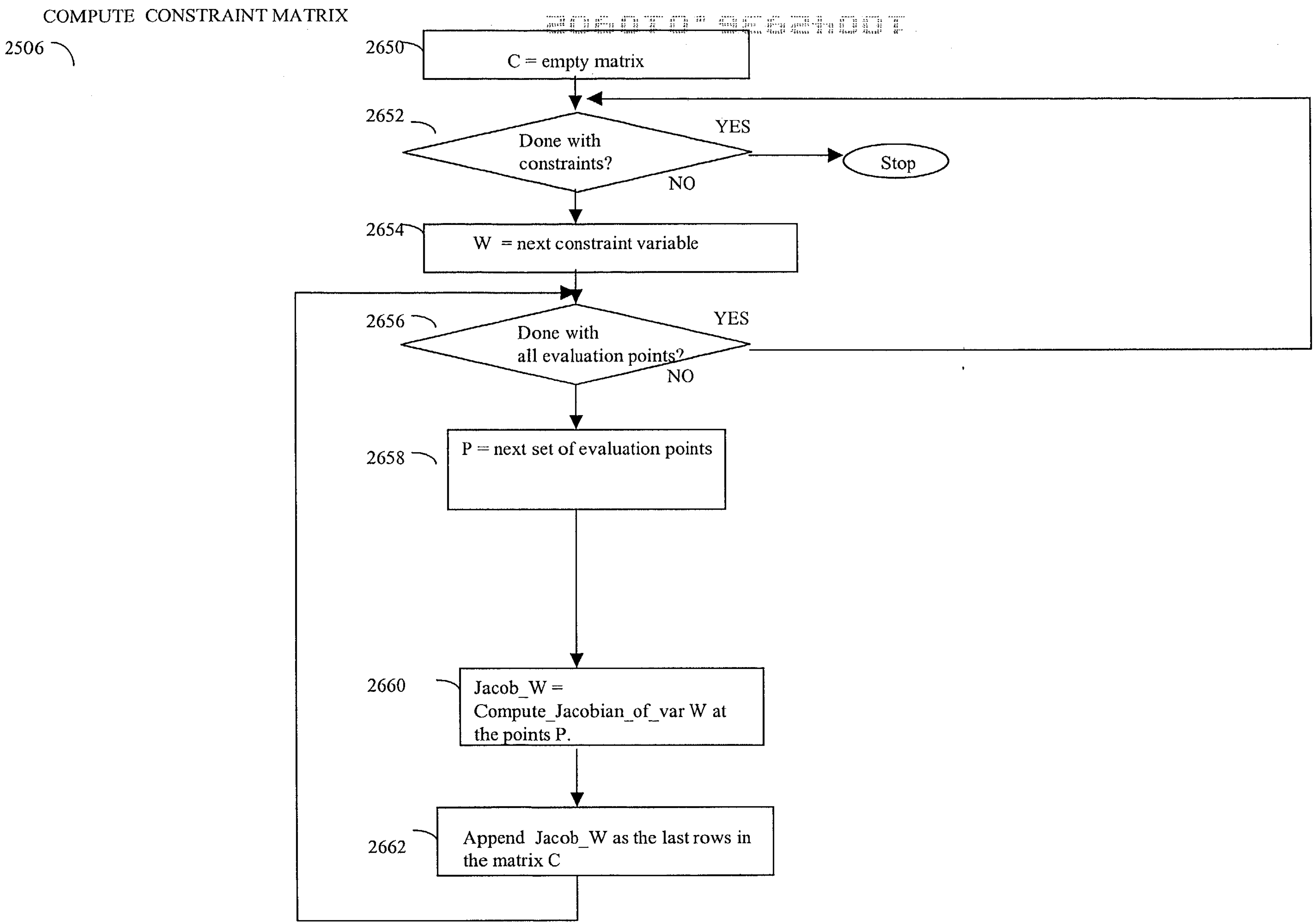

COMPUTE RESIDUAL VECTOR
2504
2600
i = 0
2602
Done with all weak terms?
YES
NO
Form residual vector by adding residual vectors of all weak terms.
2616
Stop
2604
current_weak_term = next weak term
sum = 0
2606
Done with all evaluation points?
YES
NO
2618
i = i + 1
weak_term_residual_vector[i] =sum
2608
P = next set of evaluation points
2610
Jacob_V = Compute_Jacobian_of_var V at the points P.
2612
Val_W = Compute_values_of_var W at the points P.
2614
sum = sum + Jacob_V * Val_W * (weights of points P)

FIGURE 53



COMPUTE CONSTRAINT MATRIX
2506
2650
C = empty matrix
2652
Done with constraints?
YES
Stop
NO
2654
W = next constraint variable
2656
Done with all evaluation points?
YES
NO
2658
P = next set of evaluation points
2660
Jacob_W = Compute_Jacobian_of_var W at the points P.
2662
Append Jacob_W as the last rows in the matrix C

FIGURE 54

COMPUTE CONSTRAINT RESIDUAL VECTOR

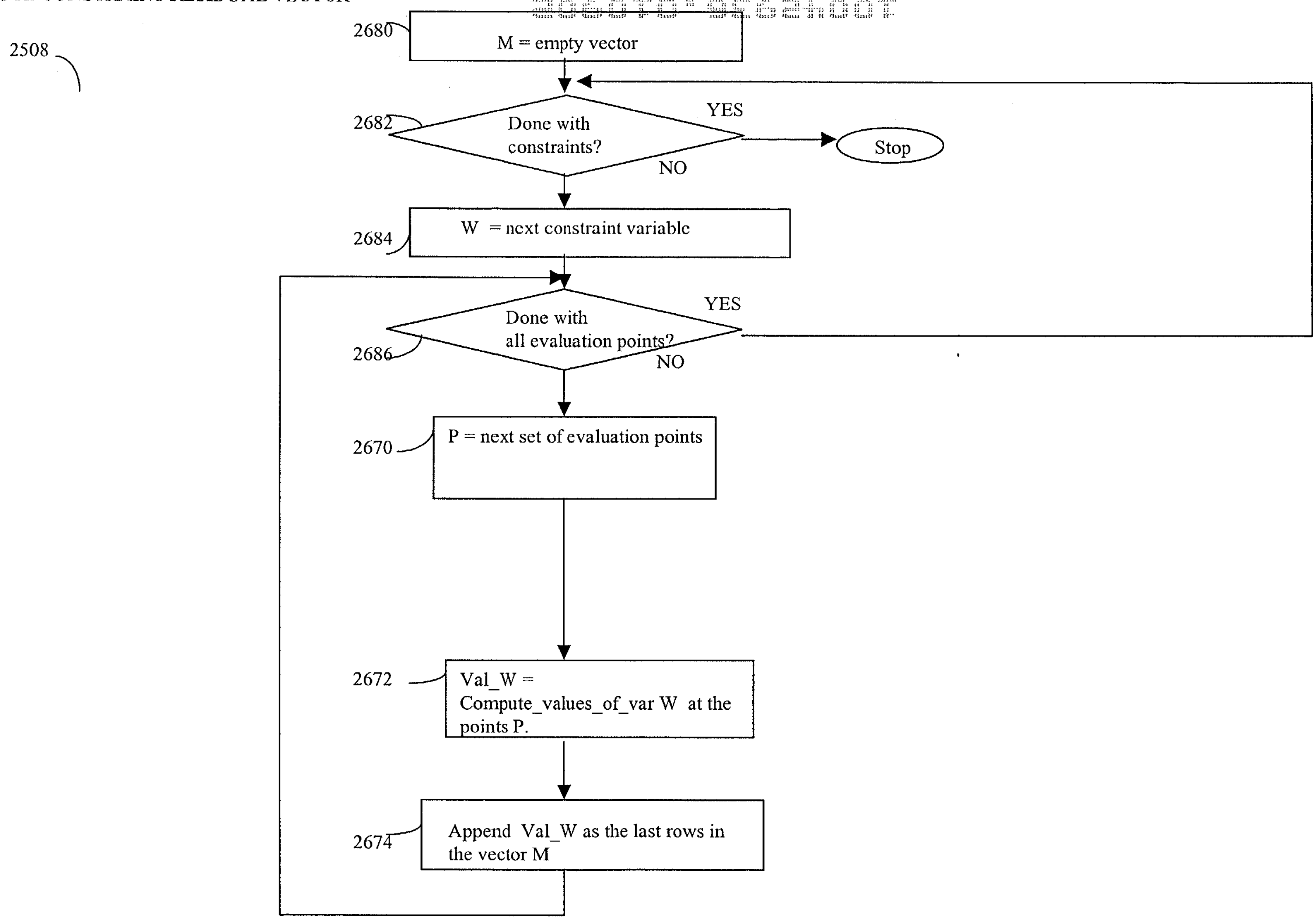


FIGURE 55A

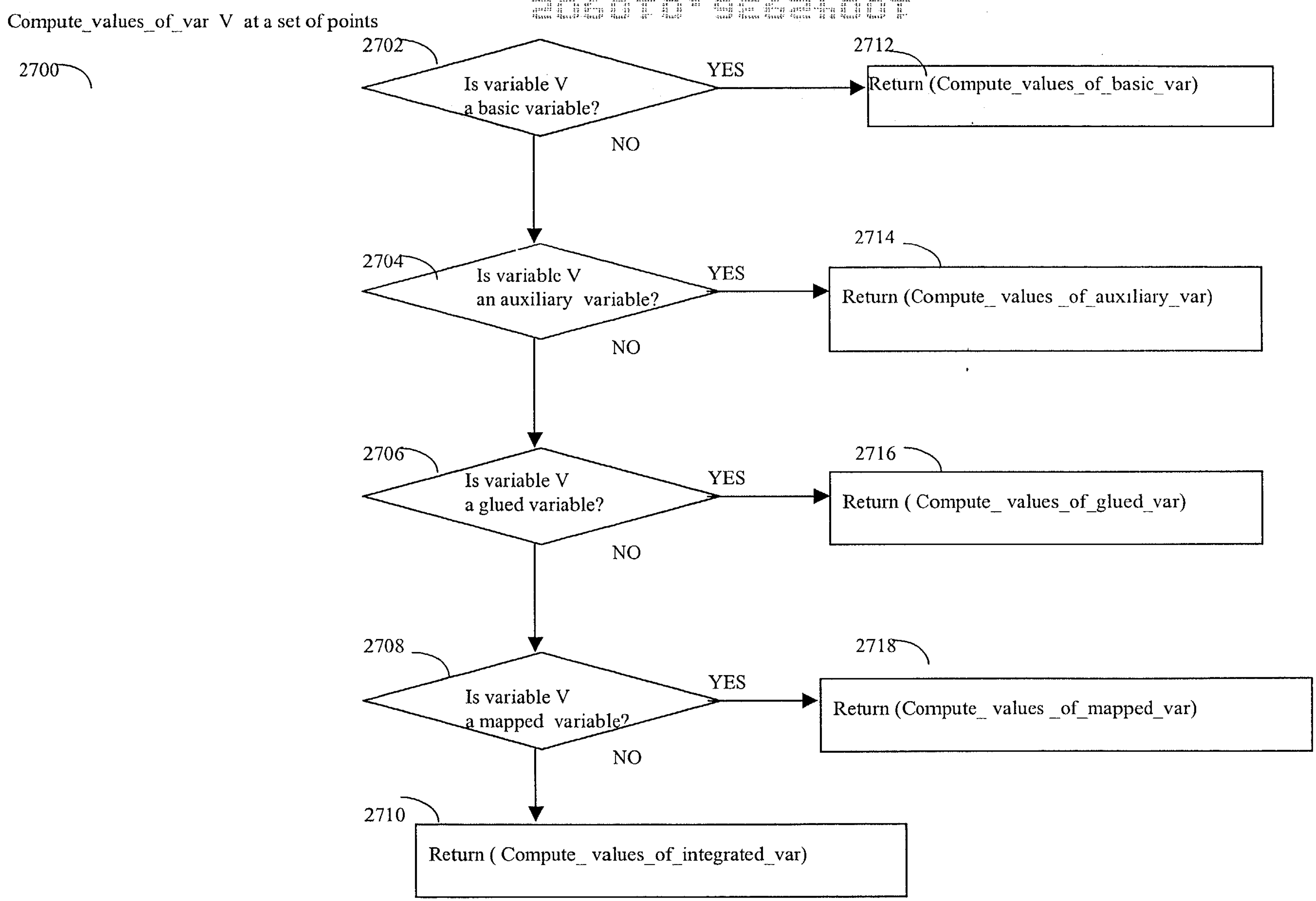

Compute_values_of_var V at a set of points
2700
2702
Is variable V a basic variable?
YES
NO
2712
Return (Compute_values_of_basic_var)
2704
Is variable V an auxiliary variable?
YES
NO
2714
Return (Compute_ values _of_auxiliary_var)
2706
Is variable V a glued variable?
YES
NO
2716
Return (Compute_ values_of_glued_var)
2708
Is variable V a mapped variable?
YES
NO
2718
Return (Compute_ values _of_mapped_var)
2710
Return (Compute_ values_of_integrated_var)

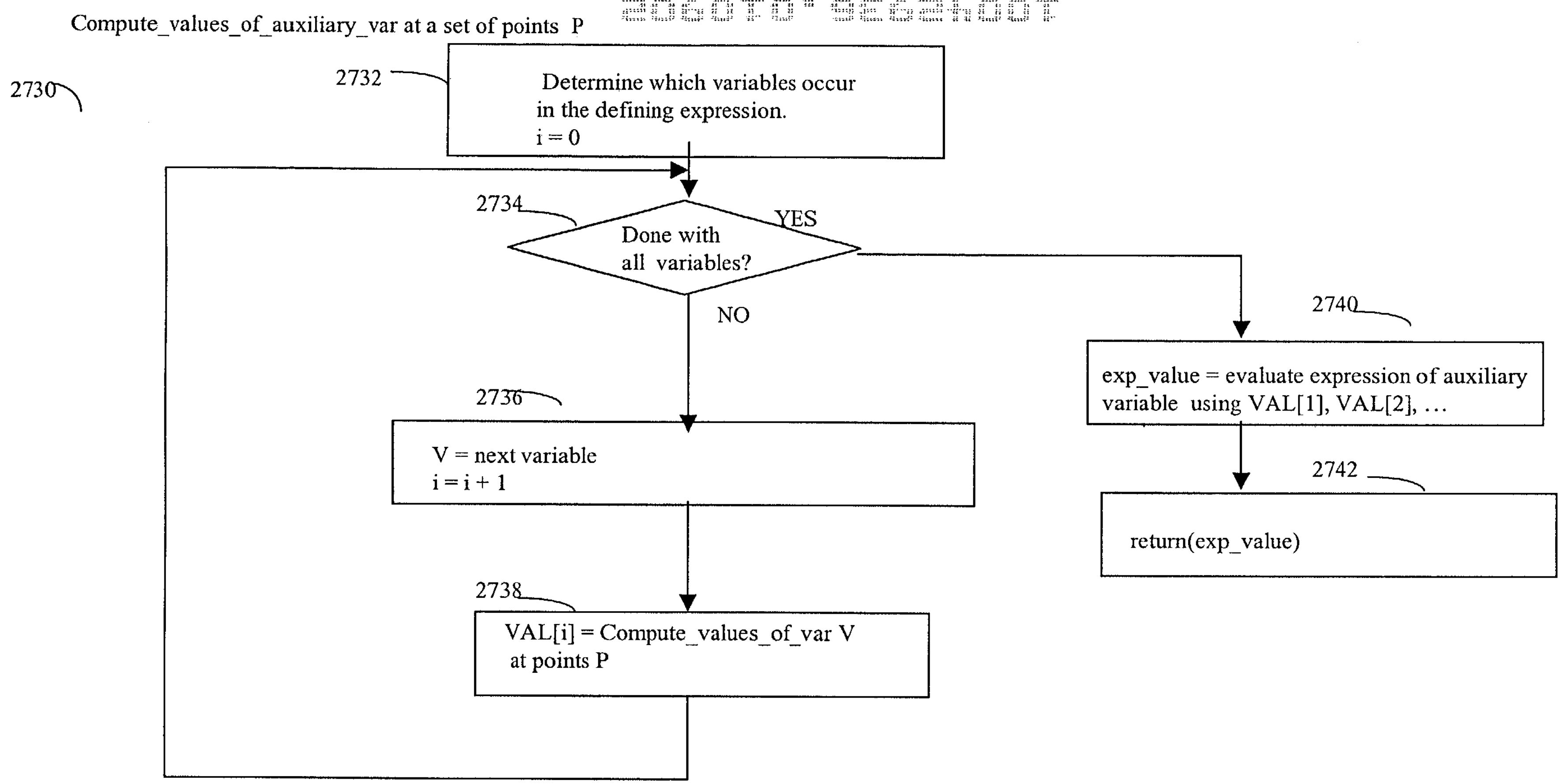
FIGURE 55B

Compute_values_of_basic_var at a set of points P

2720

Return the sums Σ Ui * Fi(pj), where the sum is taken over all indices i of the degrees of freedom, for pj in the set P.

FIGURE 55C



Compute_values_of_auxiliary_var at a set of points P
2730
2732
Determine which variables occur
in the defining expression.
i = 0
2734
Done with
all variables?
YES
NO
2736
V = next variable
i = i + 1
2738
VAL[i] = Compute_values_of_var V
at points P
2740
exp_value = evaluate expression of auxiliary
variable using VAL[1], VAL[2], …
2742
return(exp_value)

FIGURE 55D

Compute_values_of_glued_var at a set of points P

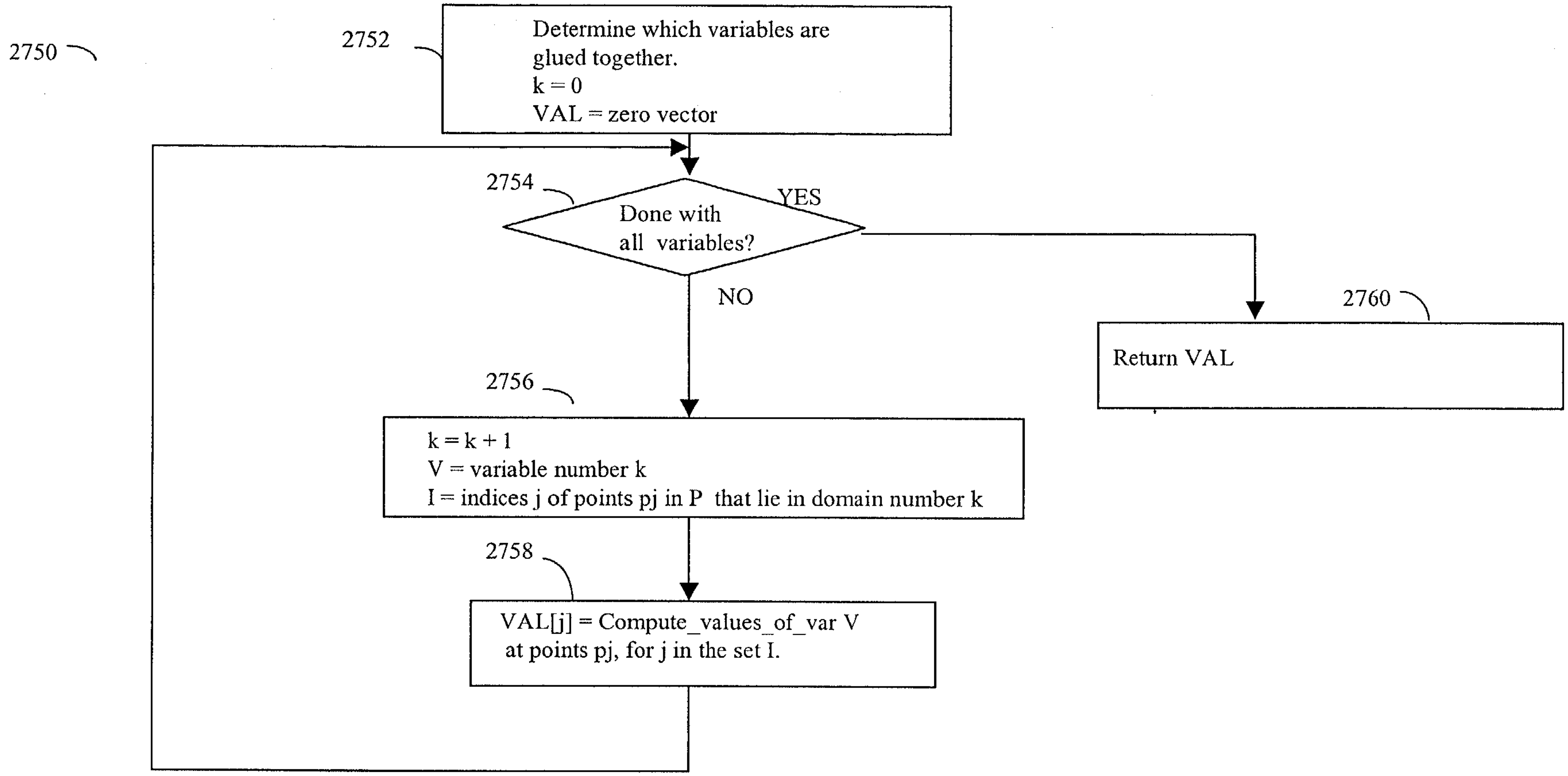


FIGURE 55E

Compute_values_of_mapped_var at a set of points P

2780

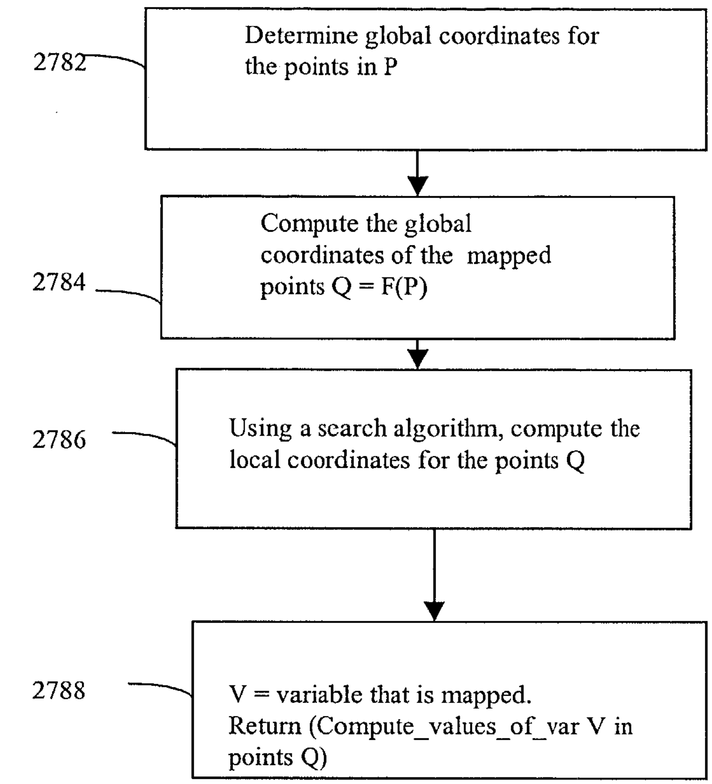


FIGURE 55F

Compute_values_of_integrated_var at a set of points P

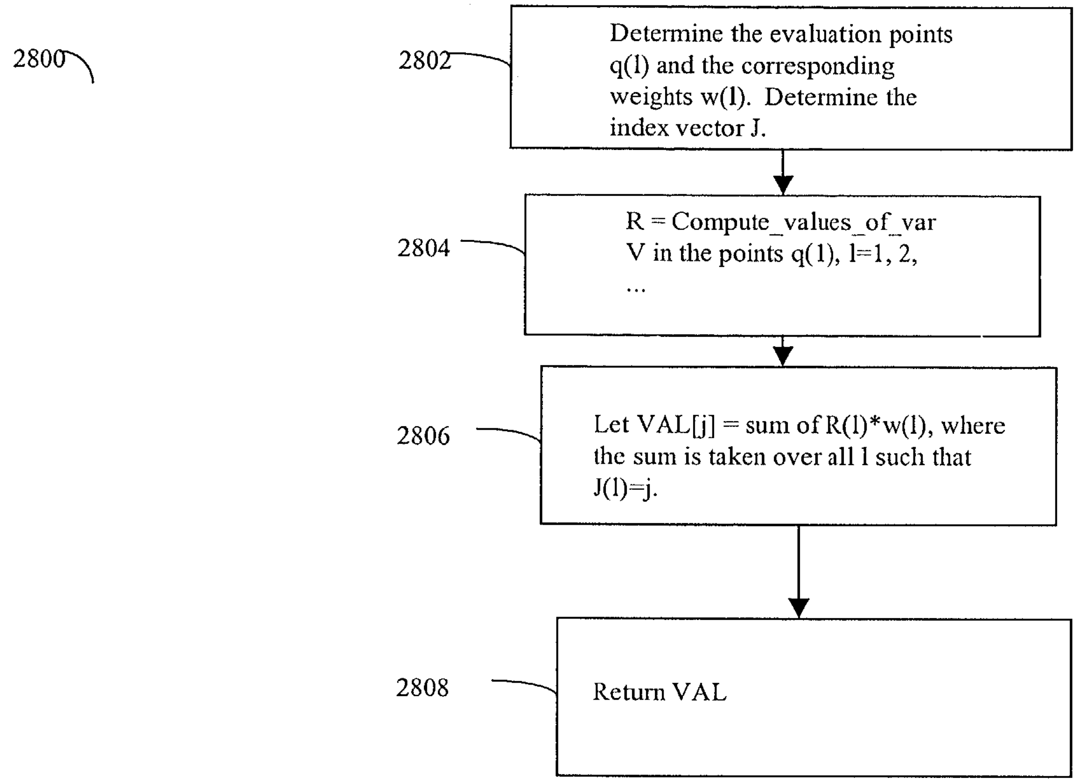


FIGURE 55G

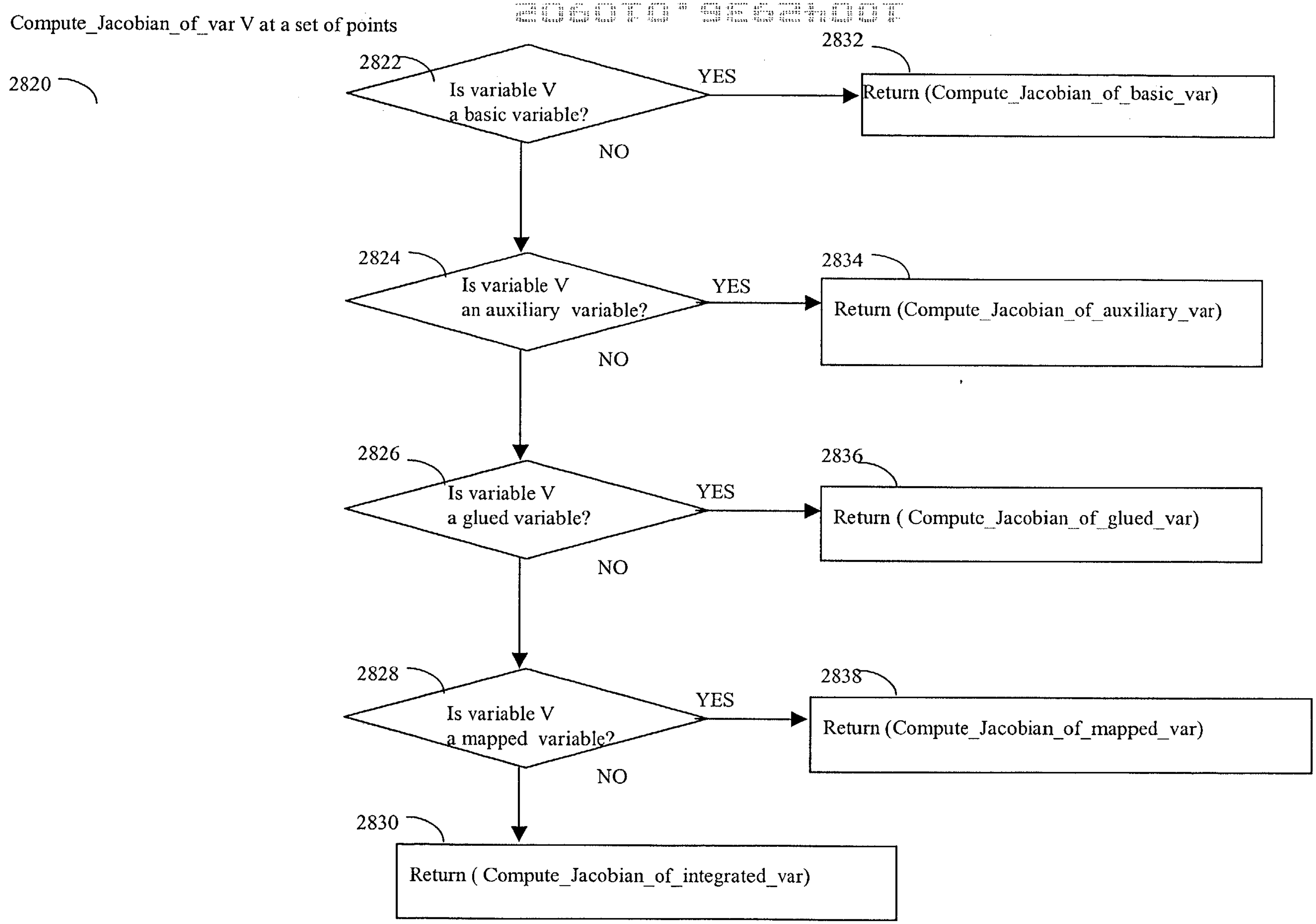

Compute_Jacobian_of_var V at a set of points
2820
2822
Is variable V a basic variable?
YES
NO
2832
Return (Compute_Jacobian_of_basic_var)
2824
Is variable V an auxiliary variable?
YES
NO
2834
Return (Compute_Jacobian_of_auxiliary_var)
2826
Is variable V a glued variable?
YES
NO
2836
Return (Compute_Jacobian_of_glued_var)
2828
Is variable V a mapped variable?
YES
NO
2838
Return (Compute_Jacobian_of_mapped_var)
2830
Return (Compute_Jacobian_of_integrated_var)

FIGURE 55H

Compute_Jacobian_of_basic_var at a set of points P

2850

2852 For the points pj in P, let the jth column of J1.DOF be the indices i of the basis functions that are nonzero on the mesh element in whch pj resides in.

2854 For the points pj in P, let the jth column of J1.VAL be the values Fi(pj), where i ranges over the contents of the jth column of J1.DOF.

2856 J1.EP = [1 , 2, ... m],
where m is the number of points in P.

2862 Return J1.

FIGURE 55I

Compute_Jacobian_of_auxiliary_var at a set of points P

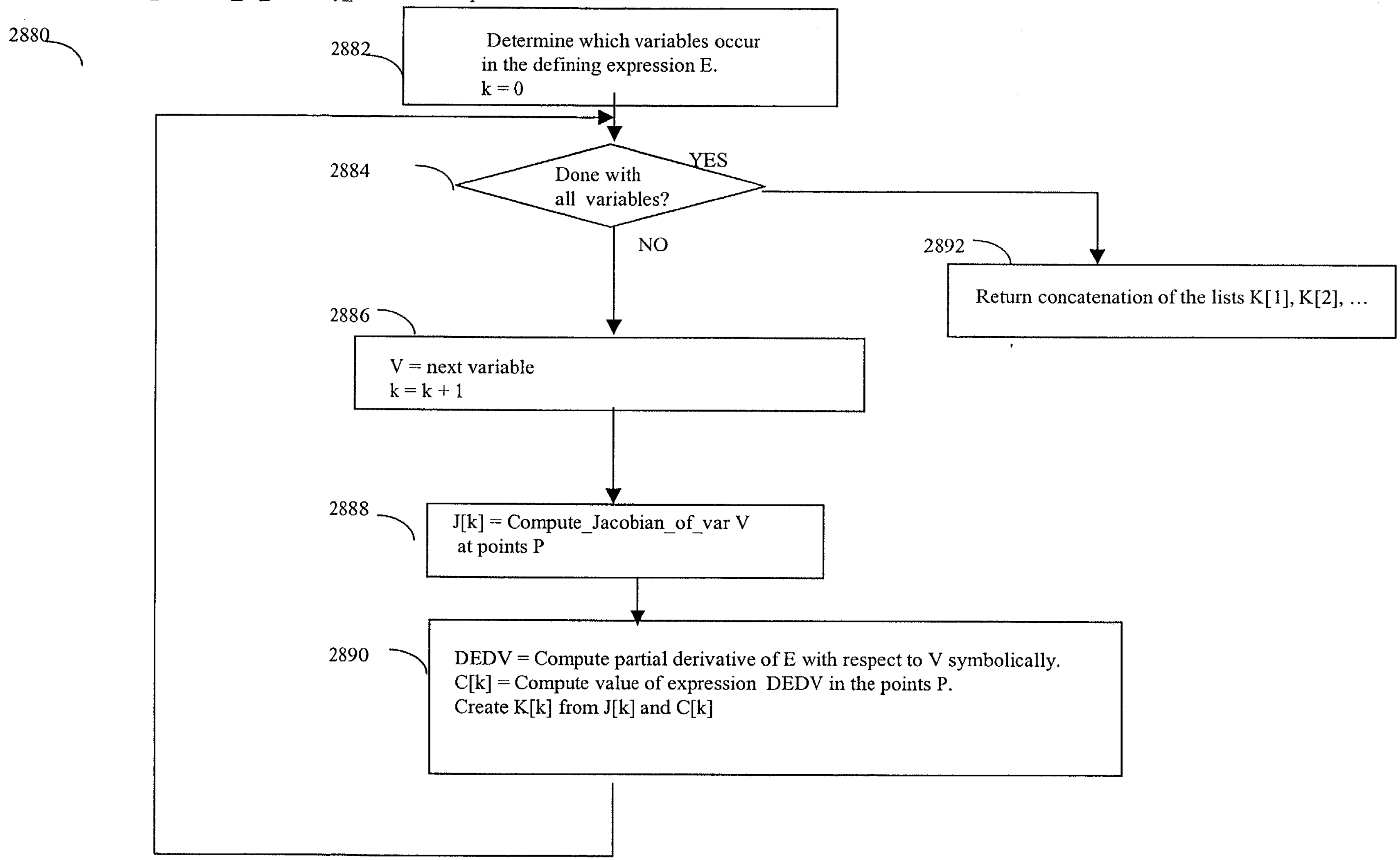


FIGURE 55J

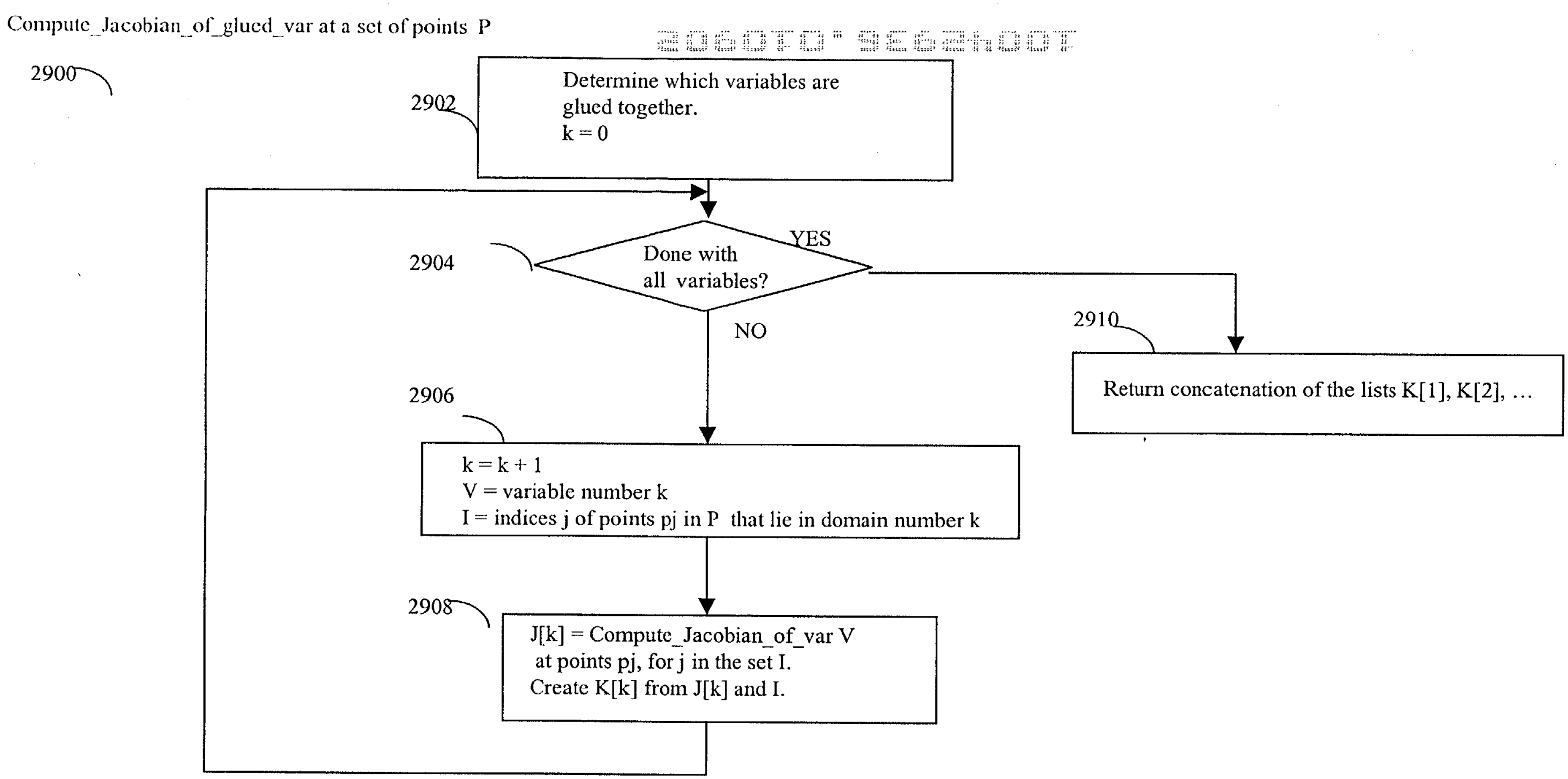

Compute_Jacobian_of_glued_var at a set of points P
2900
2902
Determine which variables are
glued together.
k = 0
2904
Done with
all variables?
YES
NO
2910
Return concatenation of the lists K[1], K[2], …
2906
k = k + 1
V = variable number k
I = indices j of points pj in P that lie in domain number k
2908
J[k] = Compute_Jacobian_of_var V
at points pj, for j in the set I.
Create K[k] from J[k] and I.

FIGURE 55K

Compute_Jacobian_of_mapped_var at a set of points P

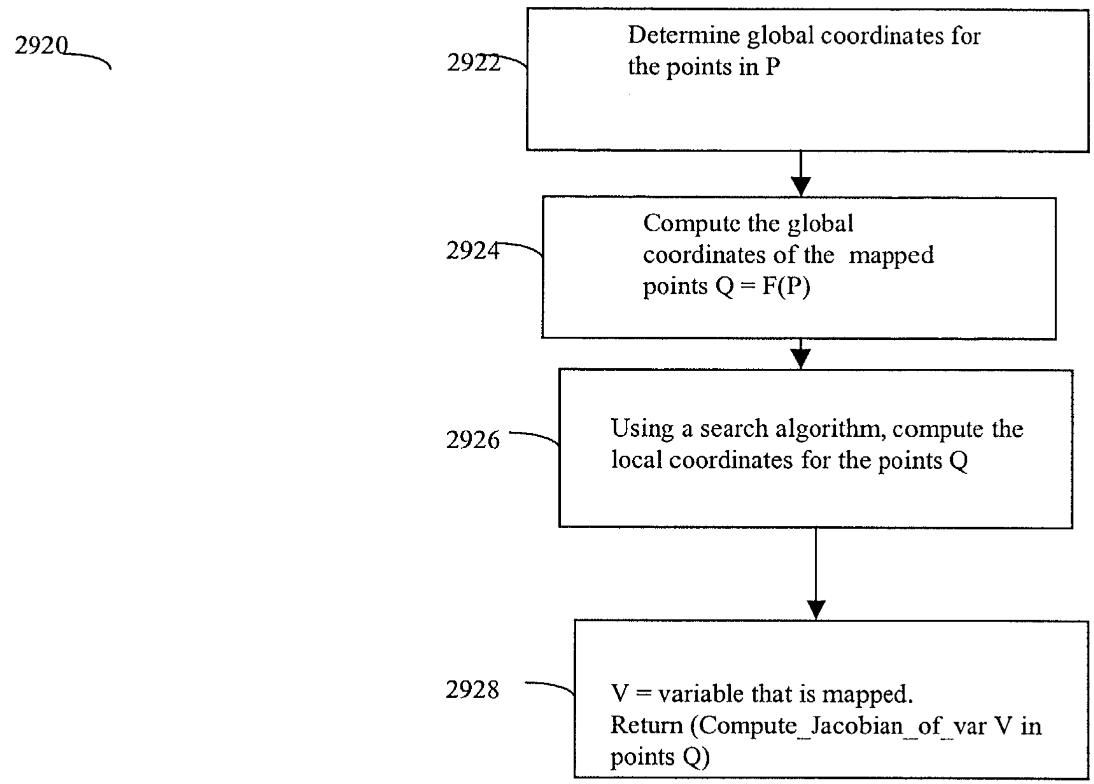


FIGURE 55L

Compute_Jacobian_of_integrated_var at a set of points P

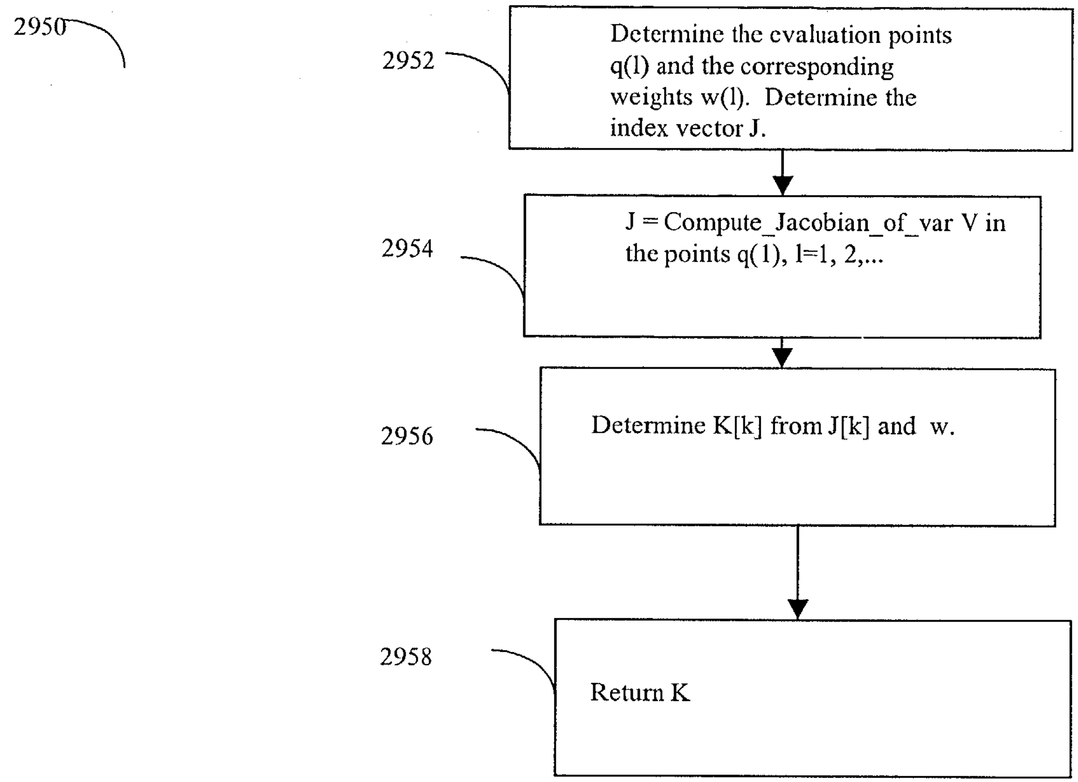


FIGURE 55M

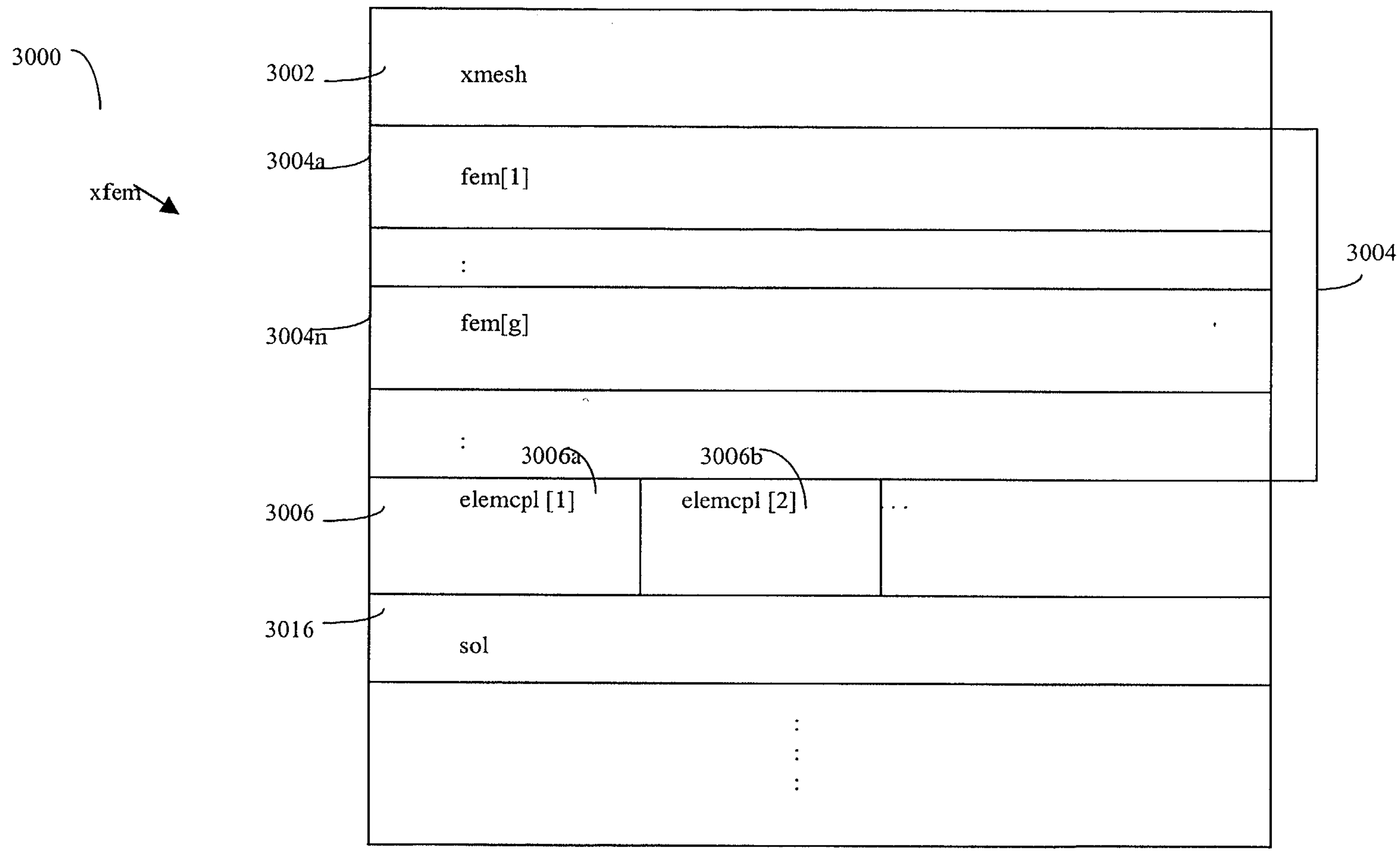

3000
xfem
3002
xmesh
3004a
fem[1]
:
3004n
fem[g]
:
3004
3006a
3006b
3006
elemcpl [1]
elemcpl [2]
...
3016
sol
:
:
:

FIGURE 56

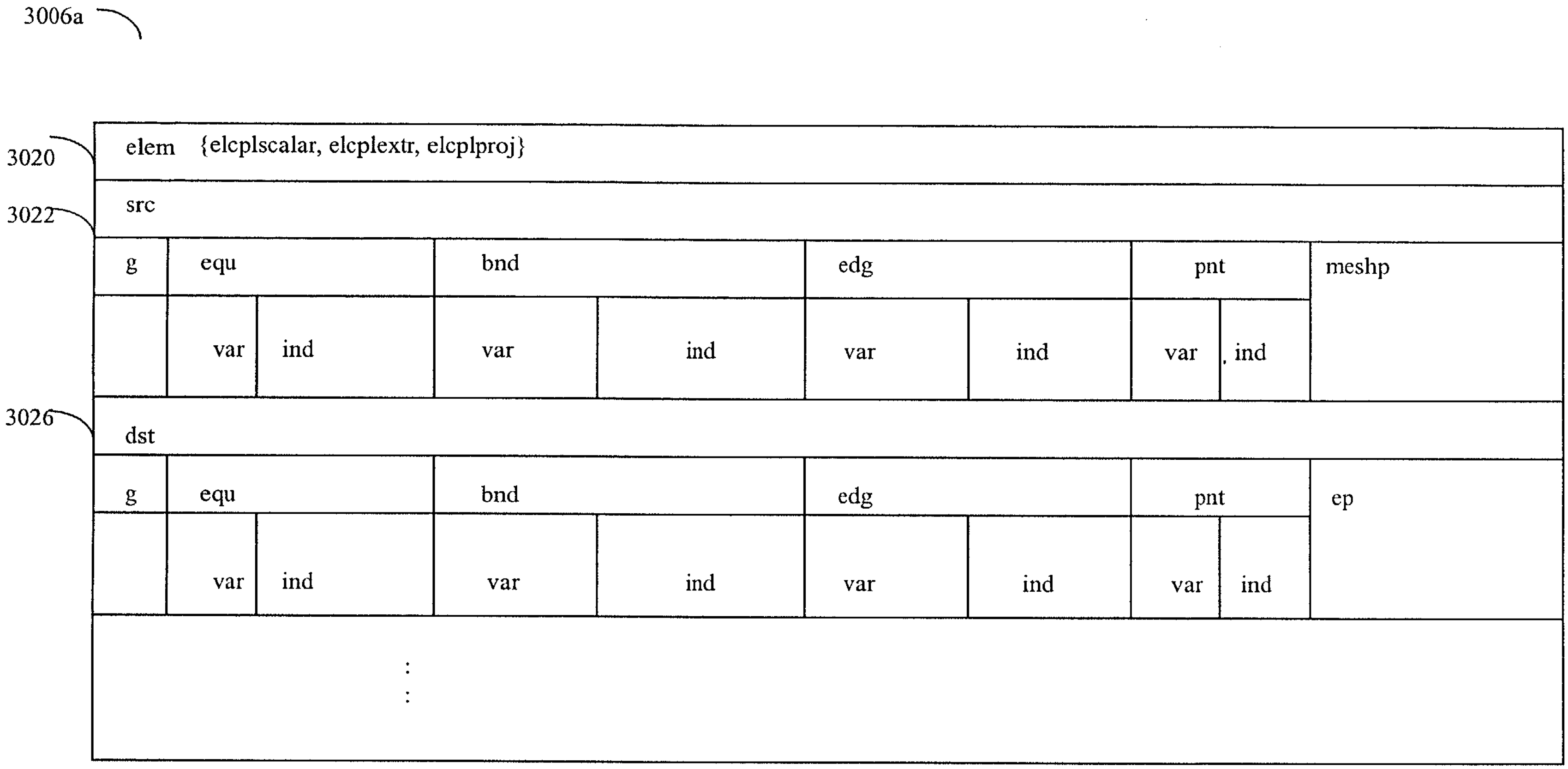

3006a
3020
3022
3026
elem {elcplscalar, elcplextr, elcplproj}
src
g
equ
bnd
edg
pnt
meshp
var
ind
var
ind
var
ind
var
ind
dst
g
equ
bnd
edg
pnt
ep
var
ind
var
ind
var
ind
var
ind

FIGURE 57

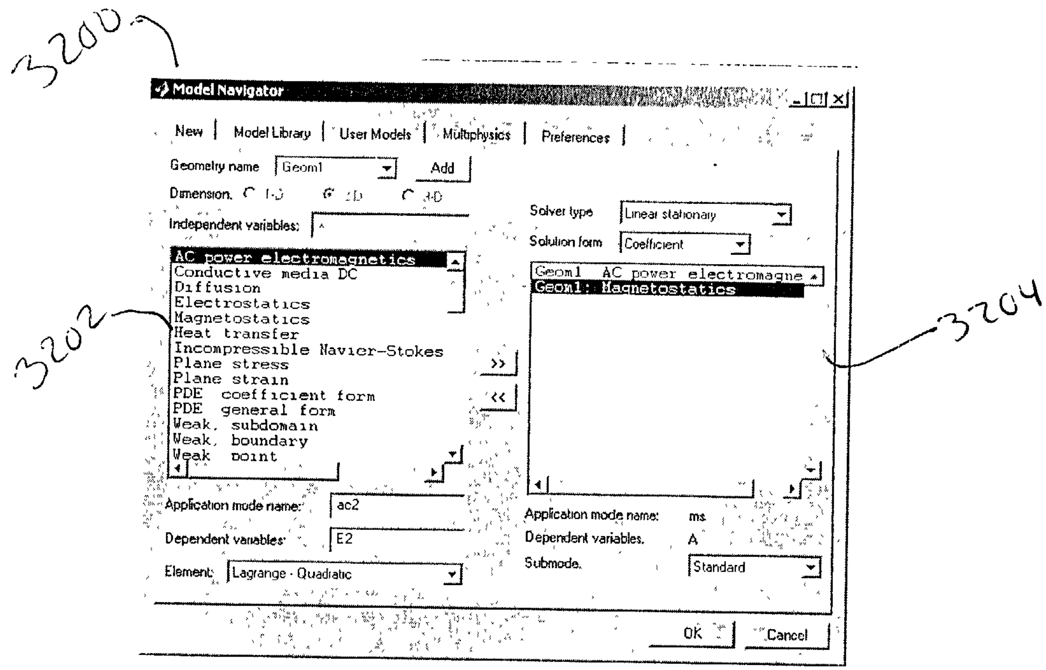

3200
3202
3204
Model Navigator
New
Model Library
User Models
Multiphysics
Preferences
Geometry name
Geom1
Add
Dimension.
Independent variables:
AC power electromagnetics
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Plane stress
Plane strain
PDE coefficient form
PDE general form
Weak, subdomain
Weak, boundary
Weak point
Solver type
Linear stationary
Solution form
Coefficient
Geom1 AC power electromagne
Geom1: Magnetostatics
Application mode name:
ac2
Dependent variables
E2
Element:
Lagrange - Quadratic
Application mode name:
ms
Dependent variables.
A
Submode.
Standard
OK
Cancel

FIGURE 58

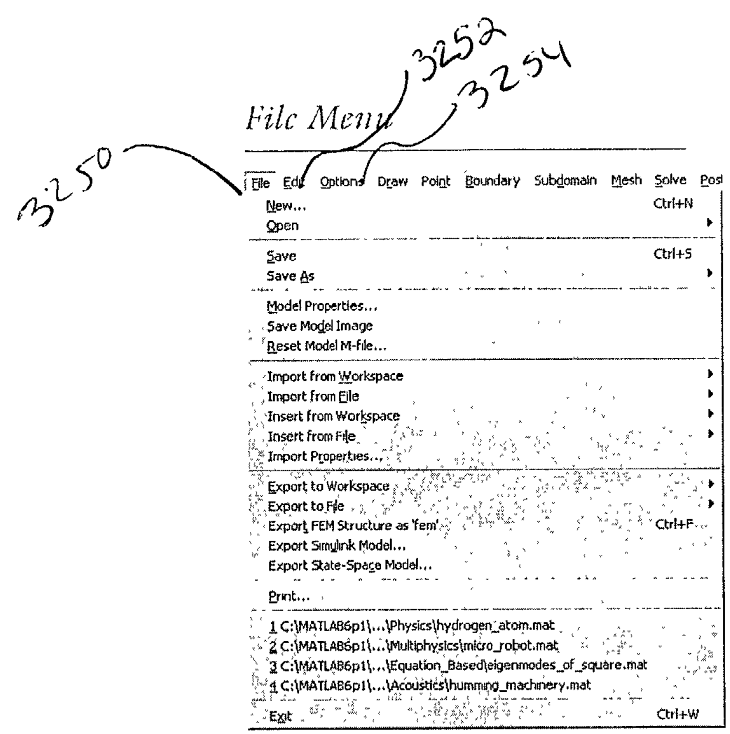
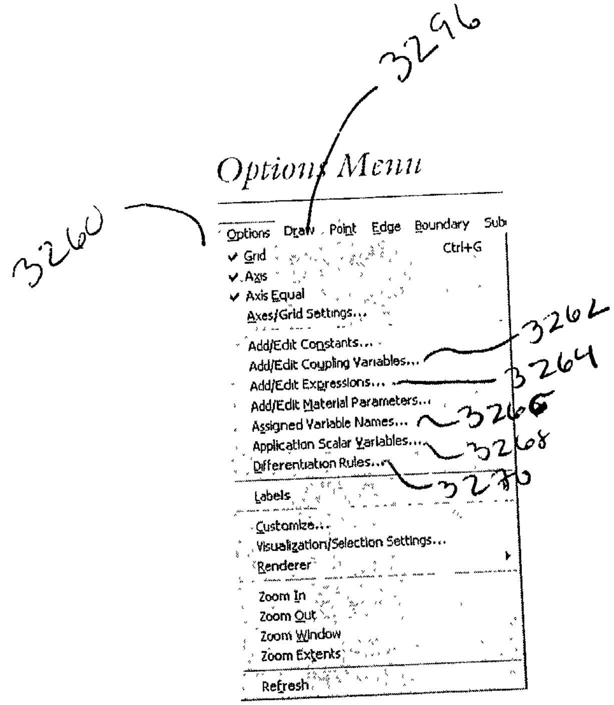

File Menu
3250
3252
3254
File Edit Options Draw Point Boundary Subdomain Mesh Solve Post
New... Ctrl+N
Open
Save Ctrl+S
Save As
Model Properties...
Save Model Image
Reset Model M-file...
Import from Workspace
Import from File
Insert from Workspace
Insert from File
Import Properties...
Export to Workspace
Export to File
Export FEM Structure as 'fem' Ctrl+F
Export Simulink Model...
Export State-Space Model...
Print...
1 C:\MATLAB6p1\...\Physics\hydrogen_atom.mat
2 C:\MATLAB6p1\...\Multiphysics\micro_robot.mat
3 C:\MATLAB6p1\...\Equation_Based\eigenmodes_of_square.mat
4 C:\MATLAB6p1\...\Acoustics\humming_machinery.mat
Exit Ctrl+W

FIGURE 59



3296
Options Menu
3260
Options
Draw
Point
Edge
Boundary
Sub
Grid
Ctrl+G
Axis
Axis Equal
Axes/Grid Settings...
Add/Edit Constants...
Add/Edit Coupling Variables...
3262
Add/Edit Expressions...
3264
Add/Edit Material Parameters...
Assigned Variable Names...
3266
Application Scalar Variables...
3268
Differentiation Rules...
3270
Labels
Customize...
Visualization/Selection Settings...
Renderer
Zoom In
Zoom Out
Zoom Window
Zoom Extents
Refresh

FIGURE 60

ADD/EDIT EXPRESSIONS...

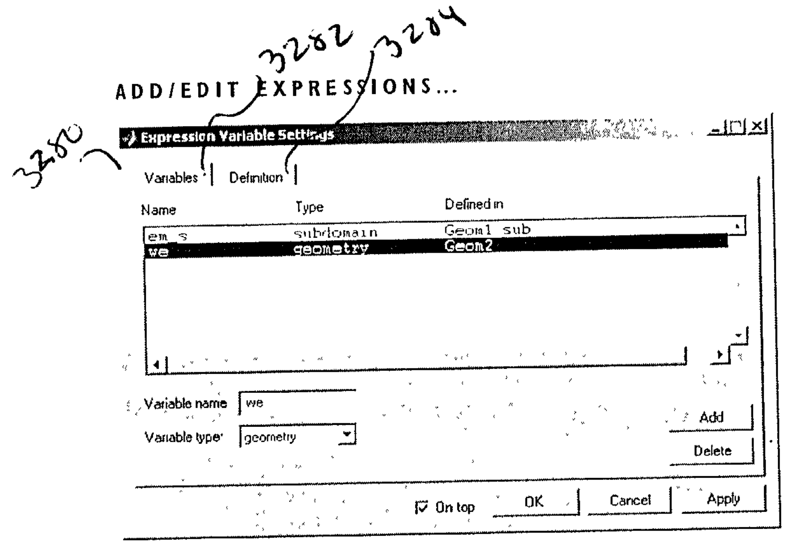


FIGURE 61

ASSIGNED VARIABLE NAMES...

3290

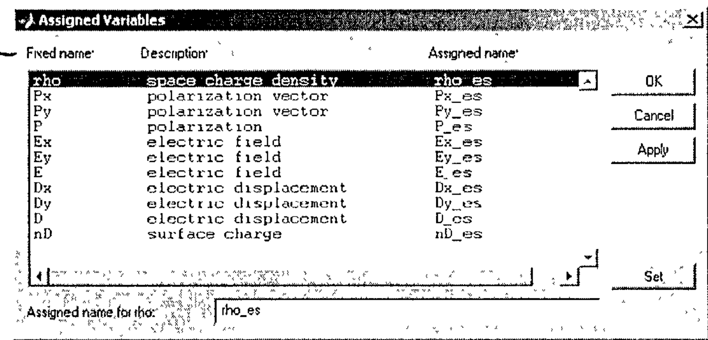


FIGURE 62

APPLICATION SCALAR VARIABLES...

3242

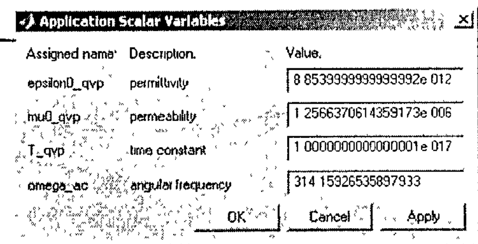


FIGURE 63

3294

DIFFERENTIATION RULES...

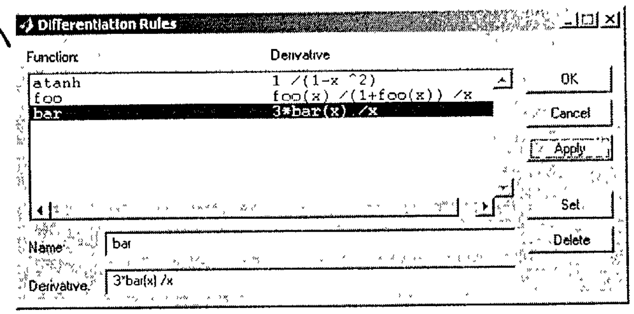


FIGURE 64

3310

Point Menu

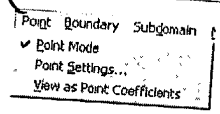


FIGURE 65

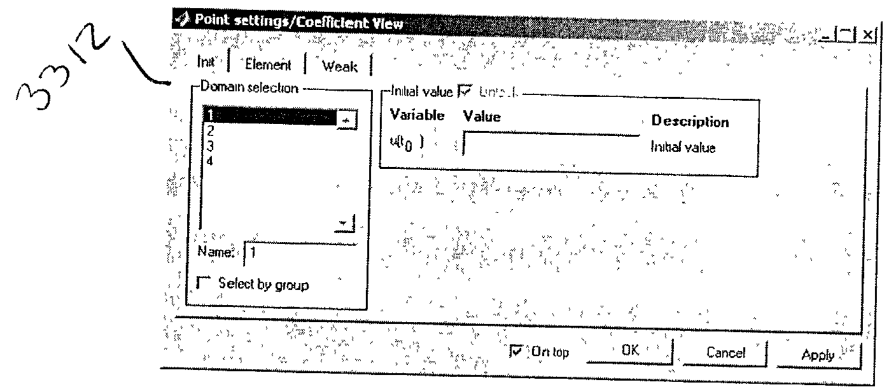

3312
Point settings/Coefficient View
Init
Element
Weak
Domain selection
1
2
3
4
Name: 1
Select by group
Initial value
Variable
Value
Description
$u(t_0)$
Initial value
On top
OK
Cancel
Apply

FIGURE 66

3314

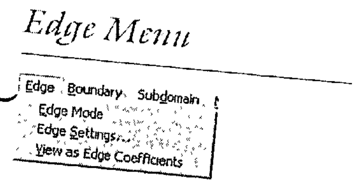


FIGURE 67

3320

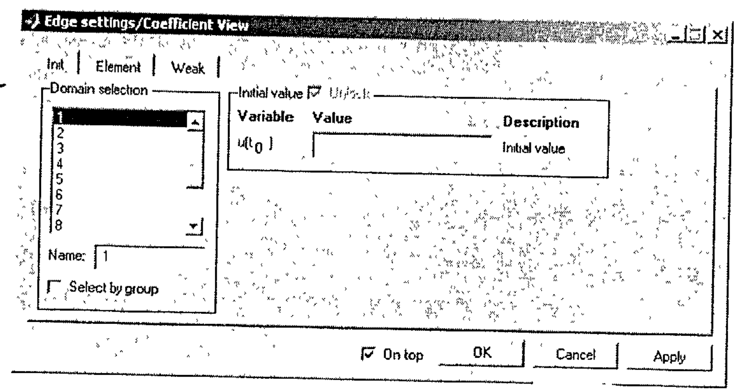


FIGURE 68

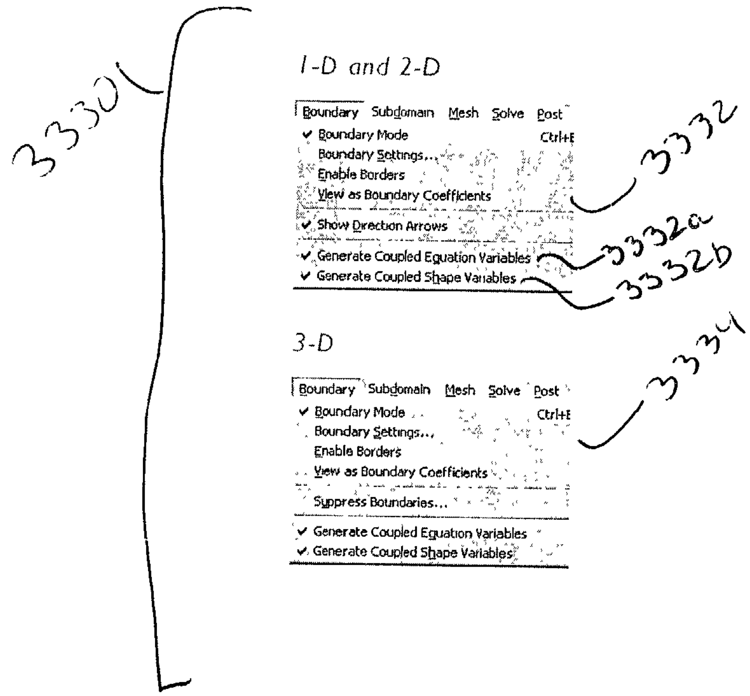

1-D and 2-D
Boundary Subdomain Mesh Solve Post
Boundary Mode Ctrl+E
Boundary Settings...
Enable Borders
View as Boundary Coefficients
Show Direction Arrows
Generate Coupled Equation Variables
Generate Coupled Shape Variables
3-D
Boundary Subdomain Mesh Solve Post
Boundary Mode Ctrl+E
Boundary Settings...
Enable Borders
View as Boundary Coefficients
Suppress Boundaries...
Generate Coupled Equation Variables
Generate Coupled Shape Variables
3330
3332
3332a
3332b
3334

FIGURE 69

3340

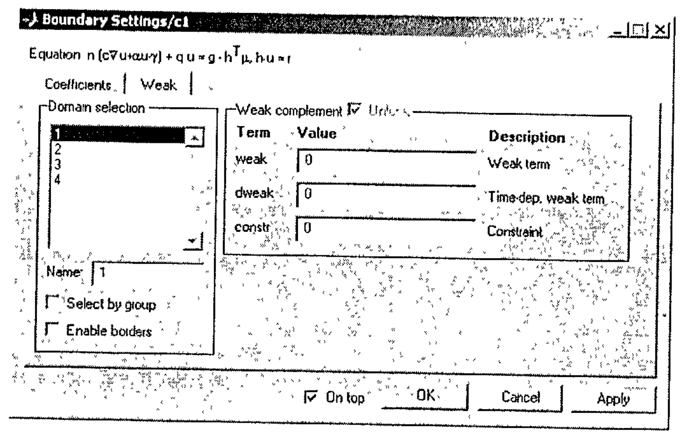


FIGURE 7D

3344

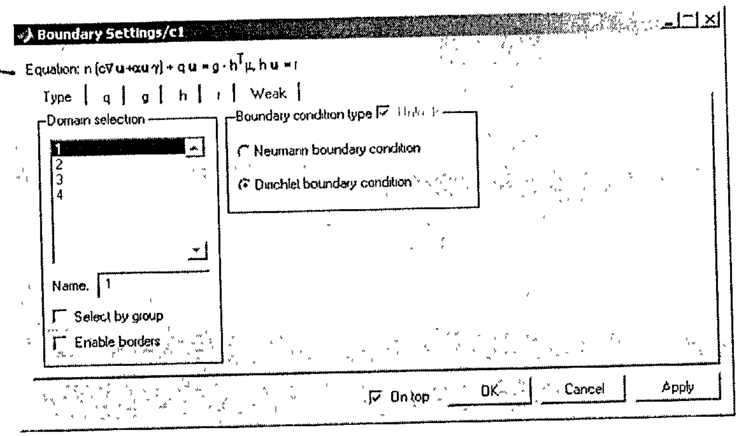


FIGURE 7L

3350

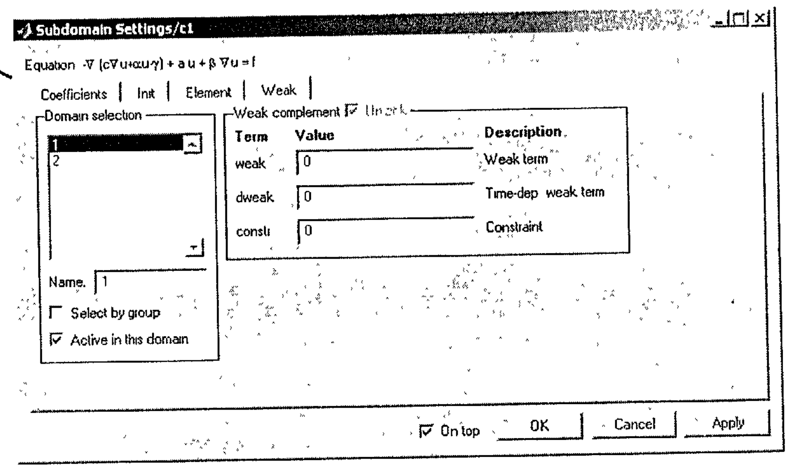


FIGURE 72

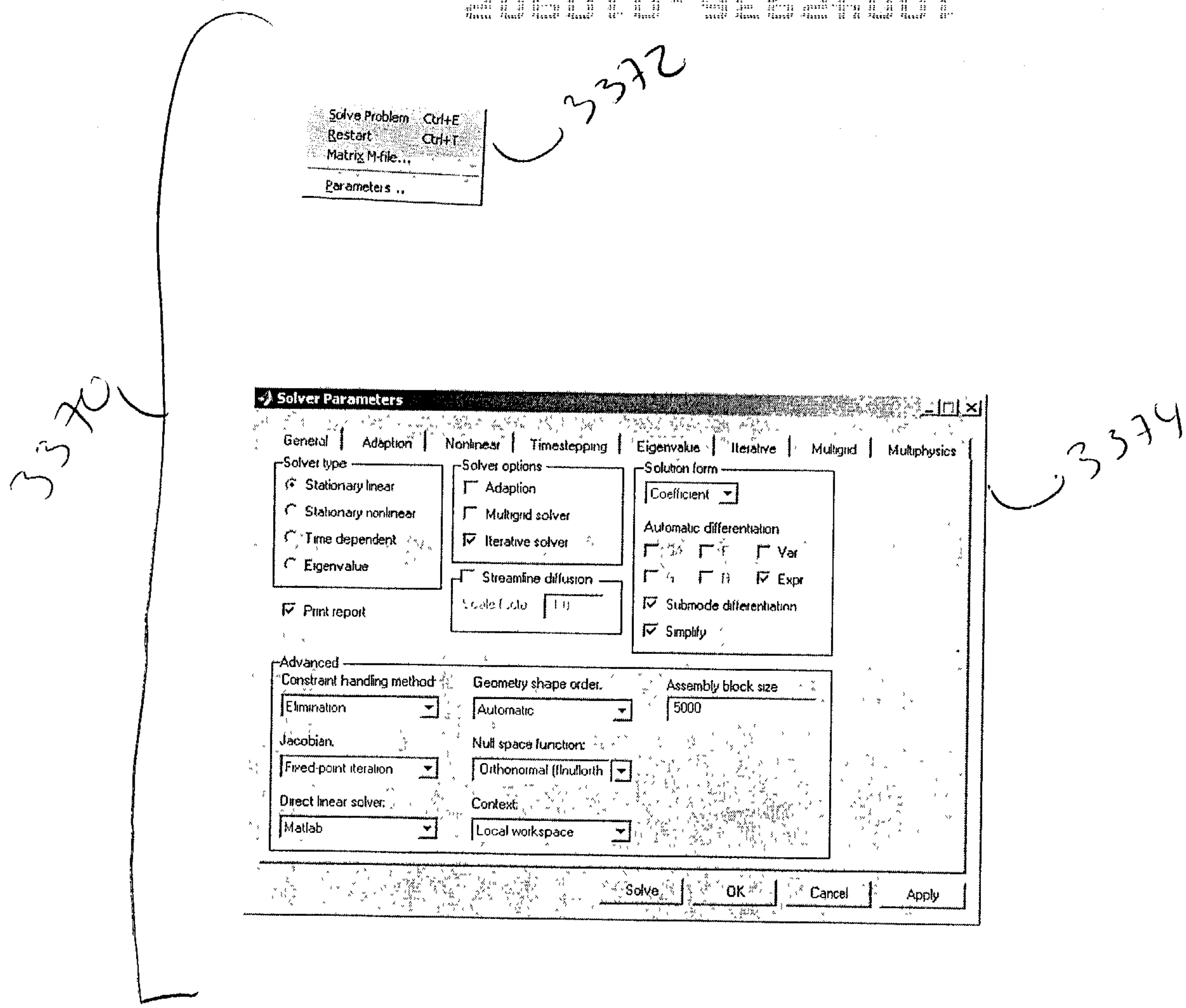

3370
3372
3374
Solve Problem Ctrl+E
Restart Ctrl+T
Matrix M-file...
Parameters ..
Solver Parameters
General
Adaption
Nonlinear
Timestepping
Eigenvalue
Iterative
Multigrid
Multiphysics
Solver type
Stationary linear
Stationary nonlinear
Time dependent
Eigenvalue
Solver options
Adaption
Multigrid solver
Iterative solver
Streamline diffusion
Solution form
Coefficient
Automatic differentiation
Var
Expr
Submode differentiation
Simplify
Print report
Advanced
Constraint handling method
Elimination
Geometry shape order.
Automatic
Assembly block size
5000
Jacobian.
Fixed-point iteration
Null space function:
Orthonormal (flnullorth
Direct linear solver.
Matlab
Context:
Local workspace
Solve
OK
Cancel
Apply

FIGURE 73

3380

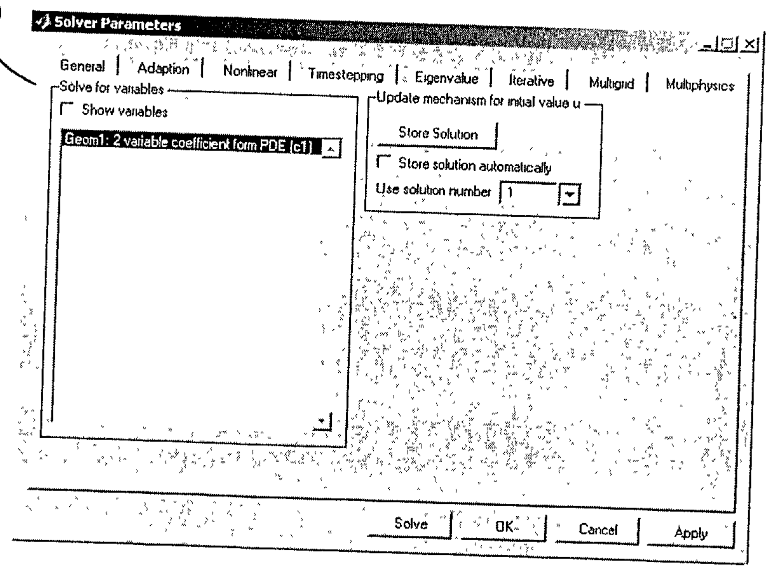


FIGURE 74

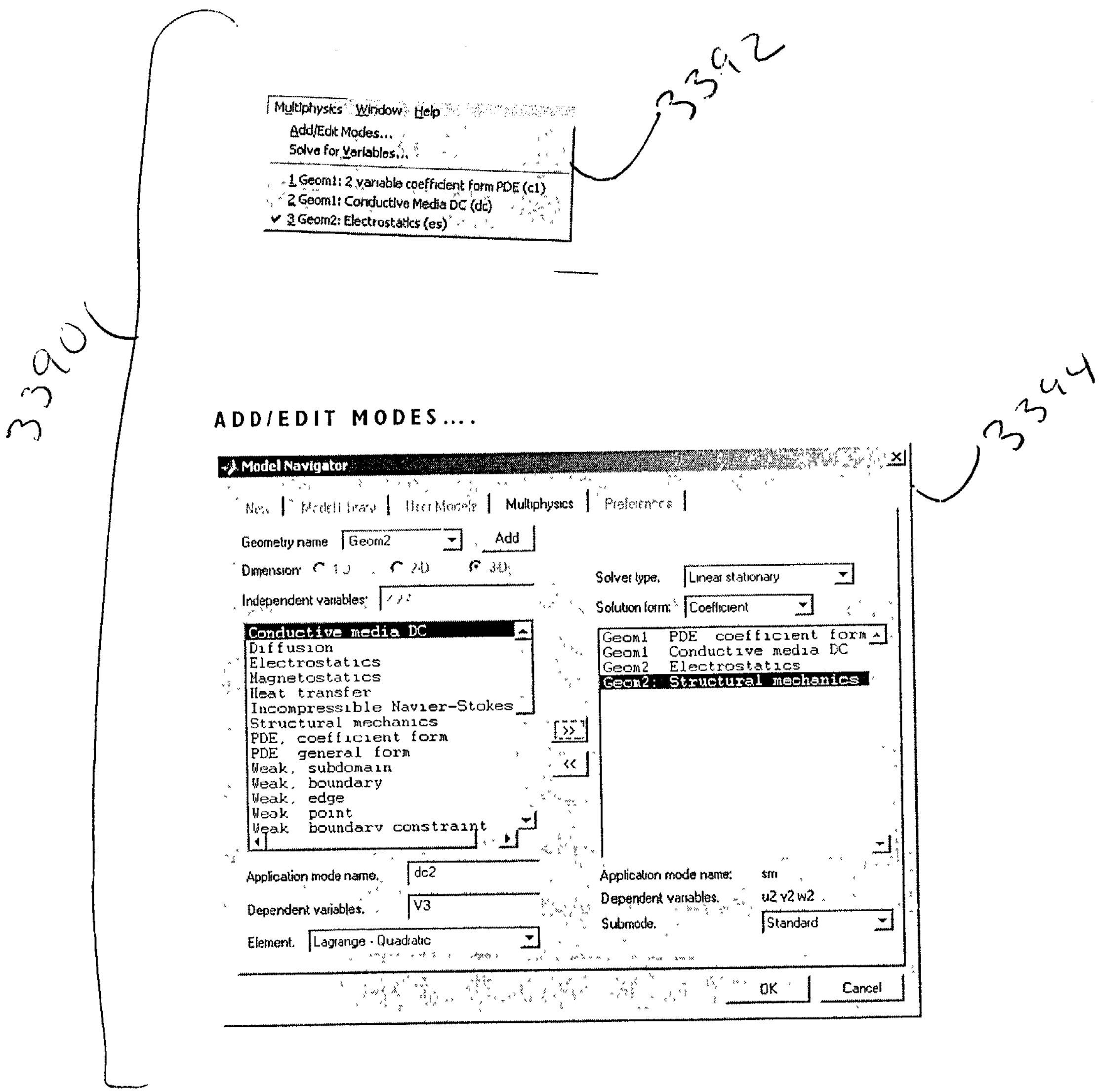

3390
3392
3394
Multiphysics
Window
Help
Add/Edit Modes...
Solve for Variables...
1 Geom1: 2 variable coefficient form PDE (c1)
2 Geom1: Conductive Media DC (dc)
3 Geom2: Electrostatics (es)
ADD/EDIT MODES....
Model Navigator
New
Multiphysics
Geometry name
Geom2
Add
Dimension
Independent variables
Conductive media DC
Diffusion
Electrostatics
Magnetostatics
Heat transfer
Incompressible Navier-Stokes
Structural mechanics
PDE, coefficient form
PDE general form
Weak, subdomain
Weak, boundary
Weak, edge
Weak point
Weak boundary constraint
Solver type.
Linear stationary
Solution form:
Coefficient
Geom1 PDE coefficient form
Geom1 Conductive media DC
Geom2 Electrostatics
Geom2: Structural mechanics
Application mode name.
dc2
Dependent variables.
V3
Element.
Lagrange - Quadratic
Application mode name:
sm
Dependent variables.
u2 v2 w2
Submode.
Standard
OK
Cancel

FIGURE 75